

Advances in Intelligent Systems and Computing 645

Elijah Blessing Rajsingh  
Jey Veerasamy  
Amir H. Alavi  
J. Dinesh Peter *Editors*

# Advances in Big Data and Cloud Computing

 Springer

# **Advances in Intelligent Systems and Computing**

Volume 645

## **Series editor**

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland  
e-mail: [kacprzyk@ibspan.waw.pl](mailto:kacprzyk@ibspan.waw.pl)

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

### *Advisory Board*

#### Chairman

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

e-mail: [nikhil@isical.ac.in](mailto:nikhil@isical.ac.in)

#### Members

Rafael Bello Perez, Universidad Central “Marta Abreu” de Las Villas, Santa Clara, Cuba

e-mail: [rbellop@uclv.edu.cu](mailto:rbellop@uclv.edu.cu)

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

e-mail: [escorchado@usal.es](mailto:escorchado@usal.es)

Hani Hagra, University of Essex, Colchester, UK

e-mail: [hani@essex.ac.uk](mailto:hani@essex.ac.uk)

László T. Kóczy, Széchenyi István University, Győr, Hungary

e-mail: [koczy@sze.hu](mailto:koczy@sze.hu)

Vladik Kreinovich, University of Texas at El Paso, El Paso, USA

e-mail: [vladik@utep.edu](mailto:vladik@utep.edu)

Chin-Teng Lin, National Chiao Tung University, Hsinchu, Taiwan

e-mail: [ctlin@mail.nctu.edu.tw](mailto:ctlin@mail.nctu.edu.tw)

Jie Lu, University of Technology, Sydney, Australia

e-mail: [Jie.Lu@uts.edu.au](mailto:Jie.Lu@uts.edu.au)

Patricia Melin, Tijuana Institute of Technology, Tijuana, Mexico

e-mail: [epmelin@hafsamx.org](mailto:epmelin@hafsamx.org)

Nadia Nedjah, State University of Rio de Janeiro, Rio de Janeiro, Brazil

e-mail: [nadia@eng.uerj.br](mailto:nadia@eng.uerj.br)

Ngoc Thanh Nguyen, Wroclaw University of Technology, Wroclaw, Poland

e-mail: [Ngoc-Thanh.Nguyen@pwr.edu.pl](mailto:Ngoc-Thanh.Nguyen@pwr.edu.pl)

Jun Wang, The Chinese University of Hong Kong, Shatin, Hong Kong

e-mail: [jwang@mae.cuhk.edu.hk](mailto:jwang@mae.cuhk.edu.hk)

More information about this series at <http://www.springer.com/series/11156>

Elijah Blessing Rajsingh  
Jey Veerasamy · Amir H. Alavi  
J. Dinesh Peter  
Editors

# Advances in Big Data and Cloud Computing

 Springer

*Editors*

Elijah Blessing Rajsingh  
Department of Computer Sciences  
Technology  
Karunya University  
Coimbatore, Tamil Nadu  
India

Amir H. Alavi  
Department of Civil and Environmental  
Engineering  
University of Missouri  
Columbia, MO  
USA

Jey Veerasamy  
Department of Computer Science, Erik  
Jonsson School of Engineering and  
Computer Science  
University of Texas at Dallas  
Richardson, TX  
USA

J. Dinesh Peter  
Department of Computer Sciences  
Technology  
Karunya University  
Coimbatore, Tamil Nadu  
India

ISSN 2194-5357                      ISSN 2194-5365 (electronic)  
Advances in Intelligent Systems and Computing  
ISBN 978-981-10-7199-7              ISBN 978-981-10-7200-0 (eBook)  
<https://doi.org/10.1007/978-981-10-7200-0>

Library of Congress Control Number: 2017957703

© Springer Nature Singapore Pte Ltd. 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. part of Springer Nature  
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

# Contents

<b>An Ontology-Based Approach for Automatic Cloud Service Monitoring and Management</b> . . . . .	1
Kirit J. Modi, Debabrata Paul Chowdhury and Sanjay Garg	
<b>Incorporating Collaborative Tagging in Social Recommender Systems</b> . . . . .	17
K. Vani and T. Swathiha	
<b>Twitter Sentimental Analysis on Fan Engagement</b> . . . . .	27
Rasika Shreedhar Bhangle and K. Sornalakshmi	
<b>A Hybrid Semantic Algorithm for Web Image Retrieval Incorporating Ontology Classification and User-Driven Query Expansion</b> . . . . .	41
Gerard Deepak and J. Sheeba Priyadarshini	
<b>Attribute Selection Based on Correlation Analysis</b> . . . . .	51
Jatin Bedi and Durga Toshniwal	
<b>Taxi Travel Time Prediction Using Ensemble-Based Random Forest and Gradient Boosting Model</b> . . . . .	63
Bharat Gupta, Shivam Awasthi, Rudraksha Gupta, Likhama Ram, Pramod Kumar, Bakshi Rohit Prasad and Sonali Agarwal	
<b>Virtual Machine Migration—A Perspective Study</b> . . . . .	79
Christina Terese Joseph, John Paul Martin, K. Chandrasekaran and A. Kandasamy	
<b>Anomaly Detection in MapReduce Using Transformation Provenance</b> . . . . .	91
Anu Mary Chacko, Jayendra Sreekar Medicherla and S. D. Madhu Kumar	
<b>Evaluation of MapReduce-Based Distributed Parallel Machine Learning Algorithms</b> . . . . .	101
Ashish Kumar Gupta, Prashant Varshney, Abhishek Kumar, Bakshi Rohit Prasad and Sonali Agarwal	

<b>TSED: Top-k Ranked Searchable Encryption for Secure Cloud Data Storage</b> . . . . .	113
B. Lydia Elizabeth, A. John Prakash and V. Rhymend Uthariaraj	
<b>An Efficient Forward Secure Authenticated Encryption Scheme with Ciphertext Authentication Based on Two Hard Problems</b> . . . . .	123
Renu Mary Daniel, Elijah Blessing Rajsingh and Salaja Silas	
<b>Clustered Queuing Model for Task Scheduling in Cloud Environment</b> . . . . .	135
Sridevi S. and Rhymend Uthariaraj V.	
<b>Static and Dynamic Analysis for Android Malware Detection</b> . . . . .	147
Krishna Sugunan, T. Gireesh Kumar and K. A. Dhanya	
<b>Performance Analysis of Statistical-Based Pixel Purity Index Algorithms for Endmember Extraction in Hyperspectral Imagery</b> . . . . .	157
S. Graceline Jasmine and V. Pattabiraman	
<b>A Multimedia Cloud Framework to Guarantee Quality of Experience (QoE) in Live Streaming</b> . . . . .	169
D. Preetha Evangeline and Anandhakumar Palanisamy	
<b>Spectral Band Subsetting for the Accurate Mining of Two Target Classes from the Remotely Sensed Hyperspectral Big Data</b> . . . . .	185
H. N. Meenakshi and P. Nagabhushan	
<b>Semantic-Based Sensitive Topic Dissemination Control Mechanism for Safe Social Networking</b> . . . . .	197
Bhuvanewari Anbalagan and C. Valliyammai	
<b>A Random Fourier Features based Streaming Algorithm for Anomaly Detection in Large Datasets</b> . . . . .	209
Deena P. Francis and Kumudha Raimond	
<b>SBKMEDA: Sorting-Based K-Median Clustering Algorithm Using Multi-Machine Technique for Big Data</b> . . . . .	219
E. Mahima Jane and E. George Dharma Prakash Raj	
<b>Cohesive Sub-network Mining in Protein Interaction Networks Using Score-Based Co-clustering with MapReduce Model (MR-CoC)</b> . . . . .	227
R. Gowri and R. Rathipriya	
<b>Design and Development of Hybridized DBSCAN-NN Approach for Location Prediction to Place Water Treatment Plant</b> . . . . .	237
Mousi Perumal and Bhuvanewari Velumani	
<b>Coupling on Method Call Metric—A Cognitive Approach</b> . . . . .	249
K. R. Martin, E. Kirubakaran and E. George Dharma Prakash Raj	

**An Intrusion Detection System Using Correlation, Prioritization and Clustering Techniques to Mitigate False Alerts** . . . . . 257  
 Andrew J. and G. Jasper W. Kathrine

**Performance Analysis of Clustering-Based Routing Protocols for Wireless Sensor Networks** . . . . . 269  
 B. Chandirika and N. K. Sakthivel

**A Secure Encryption Scheme Based on Certificateless Proxy Signature** . . . . . 277  
 K. Sudharani and P. N. K. Sakthivel

**A Two-Stage Queue Model for Context-Aware Task Scheduling in Mobile Multimedia Cloud Environments** . . . . . 287  
 Durga S, Mohan S and J. Dinesh Peter

**Degree of Match-Based Hierarchical Clustering Technique for Efficient Service Discovery** . . . . . 299  
 P. Premalatha and S. Subasree

**Providing Confidentiality for Medical Image—An Enhanced Chaotic Encryption Approach** . . . . . 309  
 M. Y. Mohamed Parvees, J. Abdul Samath and B. Parameswaran Bose

**A Novel Node Collusion Method for Isolating Sinkhole Nodes in Mobile Ad Hoc Cloud** . . . . . 319  
 Immanuel Johnraja Jebadurai, Elijah Blessing Rajsingh and Getzi Jeba Leelipushpam Paulraj

**Asymmetric Addition Chaining Cryptographic Algorithm (ACCA) for Data Security in Cloud** . . . . . 331  
 D. I. George Amalarethnam and H. M. Leena

**Improved Key Generation Scheme of RSA (IKGSR) Algorithm Based on Offline Storage for Cloud** . . . . . 341  
 P. Chinnasamy and P. Deepalakshmi

**Multi-QoS and Interference Concerned Reliable Routing in Military Information System** . . . . . 351  
 V. Vignesh and K. Premalatha

**A Secure Cloud Data Storage Combining DNA Structure and Multi-aspect Time-Integrated Cut-off Potential** . . . . . 361  
 R. Pragaladan and S. Sathappan

**Enhanced Secure Sharing of PHRs in Cloud Using Attribute-Based Encryption and Signature with Keyword Search** . . . . . 375  
 M. Lilly Florence and Dhina Suresh



**Grey Wolf Optimization-Based Big Data Analytics for Dengue  
Outbreak Prediction . . . . . 385**  
R. Lakshmi Devi and L. S. Jayashree

**Design of Smart Traffic Signal System Using Internet of Things and  
Genetic Algorithm . . . . . 395**  
P. Kuppusamy, P. Kamarajapandian, M. S. Sabari and J. Nithya

**An Innovated SIRS Model for Information Spreading . . . . . 405**  
Albin Shaji, R. V. Belfin and E. Grace Mary Kanaga

# About the Editors

**Elijah Blessing Rajsingh** is currently the Registrar of Karunya University, Coimbatore, India. He received his Ph.D. from Anna University, India in 2005. His research areas include network security, mobile computing, wireless and ad hoc networks, medical image processing, parallel and distributed computing, grid computing, and pervasive computing. He has published a number of articles in reputed journals. He is a member of IEEE, CSI, and ISTE and has served as an advisory board member for various international conferences.

**Jey Veerasamy** is Director of the Center for Computer Science Education and Outreach and a member of the teaching faculty in the Department of Computer Science, University of Texas at Dallas, USA. Prior to joining the UT Dallas in August 2010, he worked in the US wireless telecom software industry (Nortel and Samsung) for 16 years, while also teaching online courses for several colleges. Having now returned to academia to focus on teaching, he travels to India regularly to offer technical lectures and workshops and shares his US experience with Indian students.

**Amir H. Alavi** received his Ph.D. degree in Civil Infrastructure Systems from Michigan State University (MSU). He also holds a M.S. and B.S. in Civil and Geotechnical Engineering from Iran University of Science & Technology (IUST). He is currently a senior research fellow in a joint project between the University of Missouri (MU), MSU, in Cooperation with the City Digital at U+ILABS in Chicago on Development of Smart Infrastructure. He is on the editorial board of several journals and is serving as ad-hoc reviewer for many indexed journals. He is among the Google Scholar 300 most cited authors within civil engineering domain (citation > 4100 times; h-index = 34). More, he is selected as the advisory board of Universal Scientific Education and Research Network (USERN), which belongs to all top 1% scientists and the Nobel laureates in the world.

**J. Dinesh Peter** is currently working in the Department of Computer Sciences Technology at Karunya University, Coimbatore. He received his Ph.D. in Computer Science from the National Institute of Technology Calicut. His research focus areas

include Big Data, image processing, and computer vision. He has authored several publications for reputed international journals. Further, he is a member of IEEE and CSI and has served as session chair and delivered plenary speeches for various international conferences and workshops.

# An Ontology-Based Approach for Automatic Cloud Service Monitoring and Management



Kirit J. Modi, Debabrata Paul Chowdhury and Sanjay Garg

**Abstract** Cloud computing provides an efficient, on-demand, and scalable environment for the benefit of end users by offering cloud services as per service level agreement (SLA) on which both user and cloud service providers are mutually agreed. As the number of cloud users is increasing day by day, sometimes cloud service providers unable to offer service as per SLA, which results in SLA violation. To detect SLA violation and to fulfill the user requirements from the service provider, cloud services should be monitored. Cloud service monitoring plays a critical role for both the customers and service providers as monitoring status helps service provider to improve their services; at the same time, it also helps the customers to know whether they are receiving the promised QoS or not as per the SLA. Most existing cloud service monitoring frameworks are developed toward service provider side. This raises the question of correctness and fairness of monitoring mechanism; on the other hand, if monitoring is applied at user side, then it would become overhead to the clients. To manage such issues, an ontology-based Automatic Cloud Services Monitoring and Management (ACSM) approach is proposed, where cloud service monitoring and management would be performed at the cloud broker, which is an intermediate entity between the user and service provider. In this approach, when SLA violation is detected, it sends an alert to both clients and service providers and generates the status report. Based on this status report, broker automatically reschedules the tasks to reduce further SLA violation.

**Keywords** Cloud service monitoring • Service Level Agreement  
Cloud service • Ontology • Rescheduling

---

K. J. Modi (✉) • D. P. Chowdhury  
U V Patel College of Engineering, Ganpat University, Gujarat, India  
e-mail: kiritmodi@gmail.com

D. P. Chowdhury  
e-mail: debabrata130891@gmail.com

S. Garg  
Nirma University, Gujarat, India  
e-mail: gargsv@gmail.com

## 1 Introduction

In the era of Internet of Things (IoT) and cloud computing, IT resources, such as server, software, bandwidth, and network have been delivered by the service provider to customer as a service through Web known as cloud services. When hundreds of thousands of servers are connected together; then it produces massive, shared capacity for computing that can be provided through software, storage, and infrastructure. Nowadays, people make use of the services through a particular application, such as Gmail, Dropbox, or Facebook. In cloud computing, the service is acquired on an as-needed basis. When cloud service provider offers services to the customers, it is equally important to measure the quality of the service offered by the service provider. Cloud services are offered to users based on the legal agreement made between service provider and user known as Service Level Agreement (SLA). Due to economic benefits of cloud computing, all small and large organizations are moving toward the cloud-based solution [1]. Thus, SLA management is one of the biggest issues in the cloud computing environment. To detect the SLA violation and what Quality of Services (QoS) is offered by the service provider, monitoring of the cloud services needs to be performed. Cloud service monitoring plays a critical role for both the user and service providers in the sense that the monitoring status helps service provider to improve their services at the same time helps the customer to know whether they are receiving the promised QoS or not as per the SLA. There are several commercial and open-source cloud service monitoring tools in the usage, but all of them are service-provider-specific so they create the question of unfairness because monitoring is performed by the service provider side. This motivates us to design and develop a fair cloud service monitoring and management system.

**Contribution:** In this paper, an ontology-based Automatic Cloud Services Monitoring and Management (ACSMM) approach is proposed, in which cloud service monitoring and management is applied at cloud broker level using SLA and ontology. The term automatic defines the ability to monitor and manage the cloud services without any kind of human interference during the process. We develop a SLA ontology model for the semantic description of the QoS parameters. Our approach automatically monitors the cloud services and sends alerts, when SLA is violated and automatically takes reactive actions to reduce the further SLA violation.

## 2 Preliminary Concepts

In this section, we introduce the preliminary concepts related to cloud service monitoring, QoS model and ontology to understand the present problem.

## 2.1 *Cloud Service Life Cycle*

Before seeing the cloud service monitoring, it is necessary to understand the concept of cloud service life cycle [2]. The existing software development models, such as, waterfall model or spiral model, are not suitable for the cloud environment because these existing models require more human which makes it time-consuming for both customers and service providers. As we know, the main characteristics of the cloud computing are scalability, elasticity, on-demand service; thus, the conventional software development models are not suitable for the cloud environment. As a result, cloud service life cycle [2] concept is introduced, which consists of following five phases.

- **Requirements:** In the service requirements phase, the consumer specifies the technical or functional requirements and non-functional requirements of the services, that they want to consume. As a result, they issue Request for Service (RFS).
- **Service Discovery:** In the service discovery phase, the RFS generated in the previous phase is used to find the service providers that meet the technical or functional and non-functional requirements of the service.
- **Service Negotiation:** In the service negotiation phase, discussion between the service provider and customer regarding the service delivered is carried-out. Based on the discussion, the key outcome of this phase is known as service level agreement (SLA).
- **Service Composition:** Sometimes some complex requirements of customers cannot be fulfilled by single service provider. These types of requirements can be fulfilled by two or more than two service providers. Thus, two or more than two service providers are combined together to meet the complex requirements and provide a single composite service to the customers.
- **Service Consumption and Monitoring:** In this phase, the services are delivered to the consumers based on the SLA. After the services are provided to the consumer, it is necessary to regularly monitor the status of delivered service to check whether delivered services meet the functional and non-functional goals of the customers as specified in the SLA.

## 2.2 *SLA*

SLA [3] is a legal agreement between service provider and customer, in which services provided by the service provider are formally defined. It also specifies the action that could be taken in case of violation. In SLA, the key objectives are known as Service Level Objective (SLO). There is always confusion between the SLA and SLO. SLA is whole agreement, which includes time, location, and cost; whereas SLO contains only key objective or Key Performance Indicators (KPI),

which can be measured. The examples of the SLO are throughput, availability, response time, etc. To describe the service level agreement, Web Service Level Agreement [WSLA] [4] is used, which is based on the XML language.

### 2.3 *Ontology*

An ontology [2] is a data model that represents knowledge as a set of concepts within a domain and their relationship between these concepts. The two standards that govern the construction of the ontology are Resource Description Framework (RDF) and Web Ontology Language (OWL). In addition to these standards, ontology is made up of two main components: classes and relationships. The aim of the ontology is to understand the domain knowledge at the same time use and share that knowledge for various applications. Ontology helps to automate the various phases of cloud service life cycle. Ontology is the key component of the Semantic Web. The usage of ontologies allows meaning oriented information processing and interoperability support.

### 2.4 *QoS Model*

QoS parameters, such as availability, throughput, and response time, are considered as part of SLA in this work, which are defined as below.

- **Availability:** Availability [3] represents the idea of anywhere and anytime access to services. Availability is calculated by the formula presented as follows:

$$\text{Availability} = \frac{(\text{Committed hour} - \text{Outage hour}) * 100}{\text{Committed hour}} \quad (1)$$

- **Throughput:** Throughput [3] represents the performance of tasks performed by a computing service over a particular time period. Throughput is calculated by the formula presented as follows:

$$\text{Throughput} = \frac{\text{Number of task executed}}{\text{Execution time of all tasks} + \text{Total Delay of the all tasks}} \quad (2)$$

- **Response time:** It is the time taken by a request until the arrival of the response at the requesting interface. The response time [3] of a task can be calculated as follows:

$$\text{Response time} = \text{Finish Time of task} - \text{Submission Time of task} \quad (3)$$

The above-defined preliminary concepts are applied by us to design and develop the cloud service monitoring and management system. To understand the importance of cloud service monitoring, we have presented the related work carried out by various researchers in the following section.

### 3 Related Work

In this section, we present the work related to cloud service monitoring published by various researchers by highlighting their key contributions as follows.

Joshi et al. [2] described a process to automate each phase of the cloud service life cycle using ontology. Authors implemented the cloud storage prototype, in which they automate cloud service discovery and cloud service consumption phase, but they haven't implemented the automation in negotiation and monitoring phases. This inspires us to propose a framework that automates the monitoring phase of the cloud service life cycle. In [5], authors discussed the requirements of SLA in cloud computing in detail and also discussed the different SLA metrics in cloud computing. The existing cloud service monitoring is based on some benchmark tests, which are not so much accurate to find the performance of the cloud services. Rehman and Zia [6] proposed a cloud service monitoring framework based on user feedback. Though, this solution is reliable and accurate but it has no solution in case of SLA violation. Khandelwal et al. [7] designed lightweight, scalable cloud service monitoring framework that provides correct and up to date measurements of performance parameter of the application. In this framework, it only measures the performance parameters and does not verify it with SLA. Sahai et al. [8] proposed an automated SLA monitoring engine for the Web services. The limitation of this approach that it only monitors the SLA defined in author-specific SLA definition language. Frey et al. [3] described the SLA life cycle, where authors discussed the general and specific KPIs, which help customers in the negotiation phase during the creation of SLA. This work helps us to understand the key QoS parameters that are defined in SLA. Mohamed et al. [9] proposed a mechanism for SLA violation detection without specifying the reactive action part when SLA is violated. Vaitheki and Urmela [10] presented an algorithm of rescheduling of resources for the SLA violation reduction, which helps us to automate the reactive action when SLA violation is detected by rescheduling of task to the lightly loaded virtual machines. Singh et al. [11] implemented an automatic resource management technique called



STAR based on SLA to provide better customer satisfaction, and they also compared STAR architecture with relevant resource management technique.

From the above literature study, we have observed that the cloud services monitoring is an important task to prevent the SLA violation; as a result, the performance of the cloud services could be improved. We have seen that the existing work on cloud service monitoring and management is service provider specific in most cases, where monitoring is performed by the service provider which raises the question of fairness of SLA violation. This motivates us to propose the automatic cloud service monitoring and management framework, in which monitoring is done by an intermediate entity which is popularly known as cloud broker. The next section present the proposed framework with necessary details.

## 4 Automatic Cloud Service Monitoring and Management

In this section, we present a framework for automatic cloud service monitoring and management and an approach developed using this framework.

### 4.1 *Automatic Cloud Service Monitoring and Management Framework*

Figure 1 shows a framework for automatic cloud service monitoring and management using SLA and ontology. This framework consists of eight main components as follows:

1. *Customers*: It may be user or computer that uses the cloud services through Web portal. The cloud services provided by the service provider may be situated anywhere in the world. Customer specifies their requirements in SLA which is used to perform the monitoring.
2. *User Interface*: User interface may be a Web portal though which customers interact with the cloud broker.
3. *Service Providers*: Service providers are those, which deliver services to the customers through Web portal. The examples of different cloud providers are Google, Amazon, Microsoft Azure, Rackspace, iWeb, CloudSigma, yahoo, salesforce, IBM, etc.
4. *Cloud Broker*: It is an intermediate entity, which interacts between the service providers and customers. It monitors cloud service to check whether SLA is violated or not. If SLA is violated, then broker performs rescheduling of the task to reduce further SLA violation.
5. *Monitoring*: This entity calculates the QoS parameters availability, throughput, and response time using Eq. (1), Eq. (2), Eq. (3), respectively and compares these parameters with the SLA. If SLA is violated, it sends alerts to both service providers and customers.

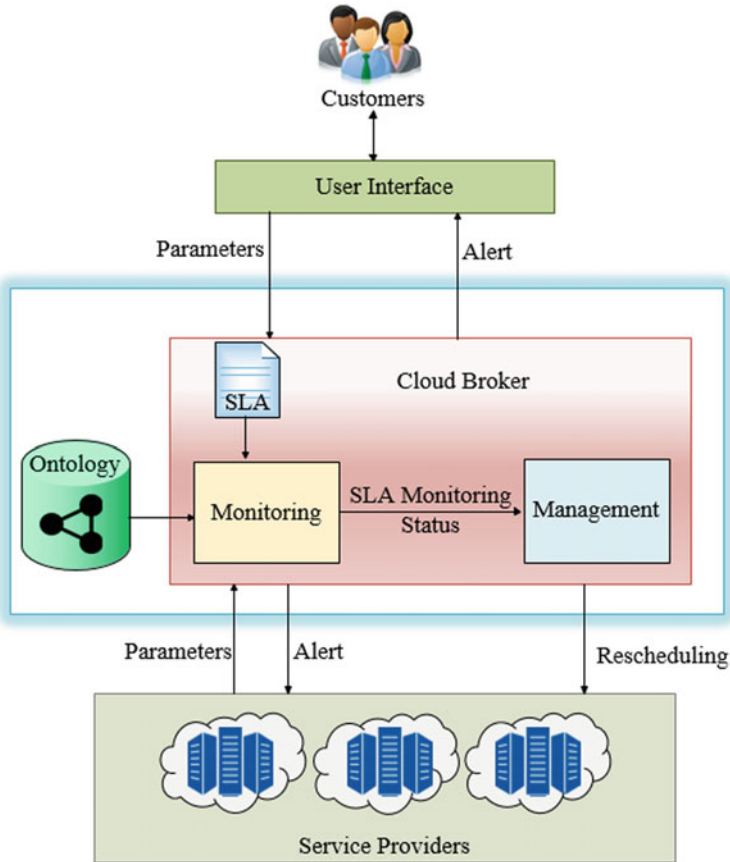


Fig. 1 Framework for automatic cloud service monitoring and management (ACSMM)

6. *Management*: This entity helps service providers to manage their resources to reduce SLA violation. When SLA is violated, this entity performs rescheduling of the task to reduce further SLA violation.
7. *SLA*: SLA is legal agreement between service provider and customers, in which services provided by the service provider is formally defined. It also specifies the action that could be taken in case of violation. This agreement is used by the monitoring entity for detection of SLA violation.
8. *Ontology*: It is a knowledge-base used by monitoring entity for SLA parameter matching in semantic manner.

In the negotiation phase of cloud service life cycle, during the creation of SLA, QoS parameters would be defined. Monitoring entity in the broker uses the SLA ontology and SLA for monitoring the cloud services. The customers and service

provider send parameters to the monitoring entity, which calculates the QoS parameters and compares with the threshold values specified in the SLA. If SLA is violated, monitoring entity sends alerts to the both customer and service provider. The monitoring entity sends the monitoring status to the management entity as an input. For a particular task, if the SLA is violated, then it reschedules the task to the other VM which is lightly loaded; thus, it reduces the further SLA violation.

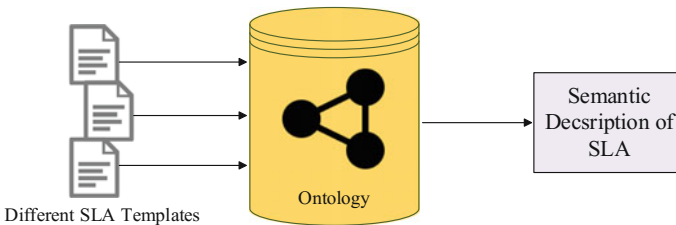
## 4.2 Ontology Model

Various SLA specification templates are proposed through which customers specify their requirements. The biggest problem with these templates is that they specify the same QoS parameters with different names. This problem can be resolved using the semantic knowledge of the SLA parameters by developing SLA ontology model.

It is important to note that the cloud platform is not providing any standards to specify the SLA parameters. To overcome the issue of heterogeneity of different SLA templates, we have developed a SLA ontology as shown in Fig. 2. This ontology stores the semantic knowledge of the SLA parameters to implement the mapping process of SLA parameters [12]. This mapping helps the monitoring entity to identify the QoS parameters defined in the SLA. Based on this information, the monitoring process is performed automatically to achieve efficiency in the presented work.

Figure 3 shows the SLA ontology, which contains the semantic information about the SLA parameters. From this information, it is clear that memory usage, memory utilization, memory consumption, storage requirement, memory requirement, and storage consumption are semantically equivalent to the storage functional requirement. Similarly, CPU, core, and processing element are semantically equivalent to the processor functional requirement.

For the non-functional requirement, we can infer that the required availability should be semantically same as the availability of the QoS parameter. Similarly, the required throughput should be semantically same as the throughput of the QoS parameter and so on.



**Fig. 2** Ontology model for SLA

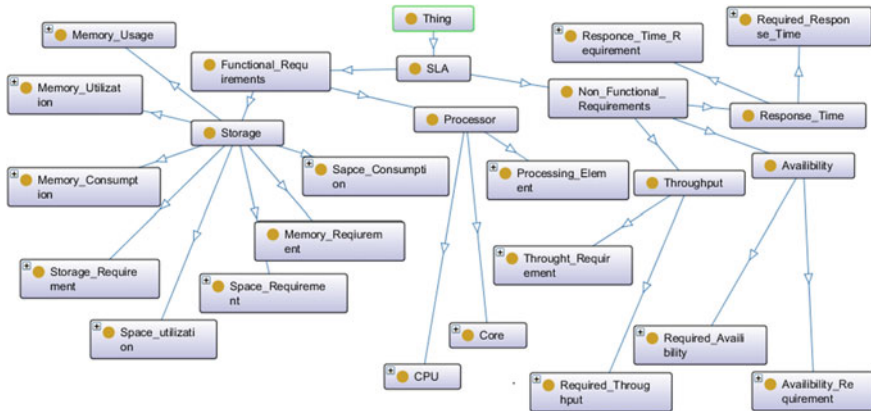


Fig. 3 SLA ontology

### 4.3 Automatic Cloud Service Monitoring and Management Approach

In this section, we present two algorithms: one is for automatic monitoring of cloud services and another is for rescheduling to manage the cloud services. In Algorithm 1, we monitor the QoS parameters of the cloud services. In case of SLA violation, an alert would be sent, and SLA violation report is delivered to the management module. The notations used in the Algorithm 1 and Algorithm 2 are defined as follows:

- *Cav (Calculated Availability)*: It is the value of the QoS parameter availability for a customer calculated by monitoring entity. This value is compared with the threshold value of the availability to check the SLA violation.
- *Crt (Calculated Response Time)*: It is the value of the QoS parameter for response time a customer calculated by monitoring entity. This value is compared with the threshold value of the response time to check the SLA violation.
- *Ctp (Calculated Throughput)*: It is the value of the QoS parameter for throughput a customer calculated by monitoring entity. This value is compared with the threshold value of the throughput to check the SLA violation.
- *Monitoring\_Status*: It monitors the status report of SLA violation for a particular time interval. This status report is used by the management entity for the rescheduling purpose.
- *Time\_Interval*: It is the time interval after which monitoring is done. It is very important to determine the appropriate monitoring interval. If we choose very large monitoring interval then we would not be able to detect all SLA violation

and if we choose very small monitoring interval then it would adversely affect the system performance. Thus, we have to choose optimum monitoring interval depending upon the consumption of cloud resources.

- *semSLA\_av* (*Threshold value for Availability*): It is semantically enabled threshold value of QoS parameter availability for a particular customer as specified in the SLA.
- *semSLA\_rt* (*Threshold value for Response*): It is semantically enabled threshold value of QoS parameter response time for a particular customer as specified in the SLA.
- *semSLA\_tp* (*Threshold value for Throughput*): It is semantically enabled threshold value of QoS parameter throughput for a particular customer as specified in the SLA.
- *sub\_time* (*Subscription Time*): It specifies the time left from the due date of the end of the subscription. In our algorithm, if the subscription time is less than 10 days, then alert would we send to both customer and service provider.
- *vm* (*Virtual Machine*): In cloud, a task is executed when it is assigned to a particular virtual machine.

---

Algorithm 1: Algorithm for SLA based Monitoring

---

Input: *semSLA\_av*, *semSLA\_rt*, *semSLA\_tp*, *sub\_time*

---

Output: *Monitoring\_Status*

---

```

1. while (MonitoringTime == true)
2.   if (Cav >= semSLA_av OR Crt <= semSLA_rt OR Ctp >= semSLA_tp OR
   sub_time < 10) then
3.     sends alert to the customer
4.     sends alert to ServiceProvider
5. Monitoring_Status ← SLA_Violation_Result
6. endif
7. wait(Time_Interval)
8. endwhile
9. return Monitoring_Status
10. exit

```

---

In the above algorithm, the semantic enabled threshold value of availability, response time, throughput, subscription time is taken as input. These threshold values are defined in SLA; based on these values, monitoring is done. The output of the algorithm is monitoring status report which is based on the SLA violation result. The SLA violation result is calculated which is based on the percentage of SLA violation which can be calculated using Eq. (4).

$$SLA_{Violation} \% = \frac{\text{Number of violated QoS Parameter}}{\text{total no QoS Parameters in SLA for a Customer}} \times 100 \quad (4)$$

First in algorithm while loop is taken, and loop is continuing until the monitoring time value become false. Then, we check the QoS parameter value with the threshold values specified in the SLA (step 2). If SLA is violated, then alert is sent

to both the client and service provider. The SLA violation result is stored in monitoring status report (step 3, 4, 5). The monitoring process is performed at regular time interval (step 7). It is very important to determine the appropriate monitoring interval. If we choose very long time interval, we will not able to take appropriate action in case of SLA violation; whereas if we choose very small monitoring interval, it will adversely affect the system performance. Thus, we have to choose optimum monitoring interval depending upon the consumption of cloud resources. At last, the algorithm returns the monitoring status report, which is used as input to the Algorithm 2. The complexity of the algorithm 1 is  $O(n)$ , where  $n$  is number of time loop is executed.

---

Algorithm 2: Algorithm for rescheduling of task

---

Input: Monitoring\_Status

---

Output: Task is assigned to vm having less load

---

```

1. for each task i in Monitoring_Status
2. if (SLA is violated for task i)
3. for each host
4. for each vm
5. Find vm having least load
6. Assign i to vm
7. endfor
8. endfor
9. endif
10. endfor
11. exit

```

---

Whenever SLA is violated, the rescheduling of task is performed as per the Algorithm 2; thus, it reduces the further SLA violation. In the Algorithm 2, the monitoring status report, which is the output of the Algorithm 1 would be the input of this algorithm. In the Algorithm 2, first we check the task whose SLA is violated (step 2). If SLA is violated algorithm find the virtual machine having least load (step 3, 4, 5). Then the selected virtual machine is assigned to that task (step 6); thus, this will reduce the further SLA violation. The complexity of the Algorithm 2 is  $O(n * m * k)$  where  $n$  is number of task,  $m$  is number of host, and  $k$  is number of host. In our approach, the monitoring is performed by Algorithm 1 after that the management of the SLA violation is performed by Algorithm 2, which is based on the monitoring status report of the task.

## 5 Experimental Setup and Results

In this section, we discuss the experimental setup and the results derived through experiments by applying our proposed work as follows:

## 5.1 Experimental Setup

We implement our framework using CloudSim 3.02 [13], which is a Java-based simulation tool for cloud environment. There are many features supported by CloudSim, i.e., network topologies, dynamic insertions of simulation entities, message passing applications, user-defined policies for resource allocation, etc. The various experiments are carried out on the machine that have 4 GB RAM, hard disk 500 GB, CPU 1.90 GHz, and Intel(R) Core(TM) i3-4030U processor. The machine is equipped with the 64 bits Windows 10 pro operating system. The tools used for the implementation are Eclipse Juno, jdk 1.8.0\_77 Apache Jena, Protégé 5.0. The SLA of each customer is specified using WSLA language.

### 5.1.1 CloudSim Configuration

In our experiment, the parameters of data center, host, virtual machine, client, and cloudlet are defined. The value of these parameters is also given here:

- *Data center*: In our experiment, we have created two data centers and the both have same configuration as given in Table 1.
- *Host*: We have created total five hosts in our experimental setup. The first two hosts are in data center 1 and remaining three hosts are in data center 2. The different configuration of host is shown in Table 2
- *Virtual Machine*: We have created six virtual machines in our experimental work. The different configuration of the virtual machines is shown in Table 3. We assign Vm1 and Vm2 to host1 and host2, respectively. We assign the two virtual machines Vm3 and Vm4 to host3. We also assign two virtual machines Vm5 and Vm6 to the host4. We define another parameter million instructions (MIPS) for virtual machines. The virtual machine having higher MIPS will have better performance for the execution.

**Table 1** Data center configuration

Arch	OS	Vmm	Time zone	Cost	Cost per memory	Cost per Storage	Cost per Bw
× 86	Linux	Xen	10.0	3.0	0.05	0.001	0.0

**Table 2** Host configuration

Host Id	Datacentre Id	No. of Pes	RAM (GB)	Bandwidth (Gbit/s)	Storage (TB)
1	1	4	1	1	1
2	1	4	1	1	1
3	2	2	2	1	1
4	2	4	4	1	1

**Table 3** VM configuration

Vm Id	MIPS	RAM (GB)	Bandwidth (Gbit/s)	No. of Pes	Vmm
1	2000	1	1	4	Xen
2	2000	1	1	4	Xen
3	1000	1	1	1	Xen
4	1000	1	1	1	Xen
5	1000	2	1	2	Xen
6	1000	2	1	2	Xen

**Table 4** Cloudlet configuration

Cloudlet Id	No. of Pes	Required availability (%)	Required throughput	Required response time (s)
1	2	95	20	250
2	2	90	10	200
3	2	100	30	300
4	2	95	25	350
5	2	98	30	500
6	2	100	25	200
8	2	98	30	350

- *Cloudlet*: We have created eight tasks as cloudlets and for that the required QoS values are specified in Table 4. These QoS values are defined in the SLA.

In our experiment, we have calculated the values of the QoS parameters of the cloudlet and then checked them with the QoS values as specified in the SLA. If SLA violation is detected, then rescheduling is performed to reduce the further SLA violation. We compare the results SLA violation percentage with management module.

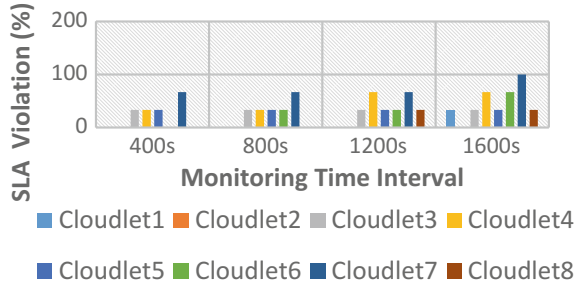
## 5.2 Experimental Results

In this section, we present the experimental results of our framework and compare the SLA violation results of with using rescheduling and without rescheduling. Figure 4 shows the SLA violation results of cloudlets without using rescheduling; thus, it means the management entity of our framework which is not included while taking the SLA violations results. Figure 5 shows the SLA violation results of cloudlets with rescheduling. We monitor the SLA parameters of the all eight cloudlets after every 400 seconds monitoring interval.

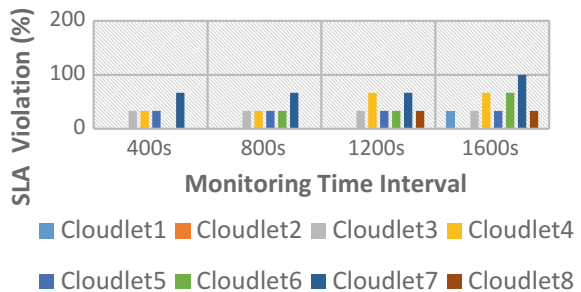
After first 400 seconds, if we compare both graphs, we get same results; as rescheduling is done, only after SLA violation is detected. After 800 seconds still



**Fig. 4** SLA violation without rescheduling



**Fig. 5** SLA violation with rescheduling



results of the both graph are same, because there is no virtual machine so load is very less. After 1200 seconds, we clearly found that the SLA violations are reduced for cloudlet 3 and cloudlet 7. After 1600 seconds, we found that the SLA violation results are reduced for the cloudlet 3, cloudlet 4, cloudlet 5, cloudlet 6, and cloudlet 8. For the other cloudlets, the results remain same. By comparing both the graphs, we can easily conclude that the SLA violation results for the cloudlets are reduced when rescheduling is applied.

### 5.3 Comparative Study with Exiting Approaches

There are several cloud service monitoring frameworks proposed, but only few of them provides mechanism to reduce the SLA violation in automatic manner. In [14], authors presented the approach to automate the QoS management, but they did not specified the detail of proposed work regarding automated QoS management. The Detecting SLA violation Infrastructure (DeSVi) architecture [15] is one of the automatic SLA violation detection architecture, which provides timely guidance depending on the consumption of resources, but this architecture is service provider oriented. A detailed survey on different cloud service monitoring tools is described in [16, 17]. As per the survey, we observed that most of tools are service provider oriented and this may raise the question of unfair monitoring of cloud services in case of SLA violation.

## 6 Conclusion and Future Work

In this paper, we proposed a framework and approach for automatic monitoring and management of the cloud services to monitor the quality of offered services using SLA ontology. When SLA violation is detected, our approach sends the alert to both service provider and user. To reduce further SLA violation, our approach automatically finds the virtual machine having light load and allocate that virtual machine to the task. We have demonstrated the experimental results derived through rescheduling and compared these with traditional (without rescheduling) approach. The results show that the automatic service monitoring and rescheduling enhance the performance of the cloud services. From the proposed work, we can specify that it is a win-win situation for both customers and service providers because monitoring is applied at broker level, so it will provide a fair SLA violation results to the users at the same time automatic rescheduling helps service providers to manage the SLA. In future work, we focus to predict the SLA violation based on the current resource conditions in cloud by applying machine learning technique. In that case, we would require the previous knowledge of SLA violation condition to predict the SLA violation. Thus, we will take action based on the prediction which will reduce the SLA violation in the cloud environment at significant level.

## References

1. Yashpalsinh, J., Modi, K.: Cloud computing-concepts, architecture and challenges. computing, electronics and electrical technologies (ICCEET), In: International Conference on. IEEE (2012)
2. Joshi, K., Yesha, Y., Finin, T.: Automating cloud services life cycle through semantic technologies. *Serv Comput. IEEE Trans.* **7**(1), 109–122 (2014)
3. Frey, S., Reich, C., Lüthje, C.: Key performance indicators for cloud computing SLAs. In: The Fifth International Conference on Emerging Network Intelligence, Emerging (2013)
4. Ludwig, H., Keller, A., Dan, A., King, R., Franck, R.: Web Service Level Agreement (WSLA) Language Specification. IBM Corporation, pp. 815–824 (2003)
5. Aljournah, E., Al-Mousawi, F., Ahmad, I., Al-Shammri, M., Al-Jady, Z.: SLA in Cloud Computing Architectures: A Comprehensive Study. *Int. J. Grid Distributed Comput.* **8**(5), 7–32 (2015)
6. Zia, et al.: A framework for user feedback based cloud service monitoring. Complex, Intelligent and Software Intensive Systems (CISIS). In: 2012 Sixth International Conference on. IEEE (2012)
7. Khandelwal, H., Kompella, R., Ramasubramanian, R.: Cloud monitoring framework. Purdue University
8. Sahai, A., Machiraju, V., Sayal, M., Jin, L., Casati, F.: Automated SLA monitoring for web services, pp. 28–41. *Management Technologies for E-Commerce and E-Business Applications*. Springer, Berlin Heidelberg (2002)
9. Mohamed, S., Yousif, A., Bakri, M.: SLA Violation detection mechanism for cloud computing. *Int. J. Comput. Appl.* **133**(6), 8–11 (2016)
10. Vaitheki, K., Urmela, S.: A SLA violation reduction technique in Cloud by Resource Rescheduling Algorithm (RRA). *Int. J. Comput. Appl. Eng. Technol.* 217–224 (2014)

11. Singh, S., Chana, I., Buyya, R.: STAR: SLA-aware autonomic management of cloud resources. *IEEE Transactions on Cloud Computing* (2017)
12. Redl, C., Breskovic, I., Brandic, I., Dustdar, S.: Automatic SLA matching and provider selection in grid and cloud computing markets. In: *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. IEEE Computer Society (2012)
13. Calheiros, R., Ranjan, R., Beloglazov, A., Rose, C., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithm. *Soft. Pract. Exp.* **41**(1), 23–50 (2011)
14. Alhamazani, K., Ranjan, R., Rabbhi, F., Wang, L., Mitra, K.: Cloud monitoring for optimizing the QoS of hosted applications. In: *IEEE 4th International Conference on IEEE* (2012)
15. Emeakaroha, V., Netto, M., Cleheiros, R., Brandic, I., Buyya, R., Rose, C.: Towards autonomic detection of SLA violations in Cloud infrastructures. *Fut. Gen. Comput. Syst.* **28**(7), 1017–1029 (2002)
16. Aceto, G., Botta, A., Donato, W., Pescape, A.: Cloud monitoring: a survey. *Comput. Netw.* **57**(9), 2093–2115 (2013)
17. Alhamazani K., Ranjan, R., Mitra, K., Rabhi, S., Khan, S., Guabtni, A.: An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. *Computing*, **97**(4), 357–377 (2015)

# Incorporating Collaborative Tagging in Social Recommender Systems



K. Vani and T. Swathiha

**Abstract** Recommender systems play a major role in recent times to help online users in finding the relevant items. Traditional recommender systems have been analysed immensely, but they ignore information like social friendships, tags which when incorporated in recommendation can improve its accuracy. With the advent of social networking sites, study of social recommender systems has become active. Most of the users ask their friends for recommendation. But not all friends have similar taste as that of the user, and different group of friends contribute to different recommendation tasks. So this paper proposes an approach to identify different group of friends by grouping users based on items and retains the personal interest of experienced by incorporating individual-based regularization in basic matrix factorization. Information like ratings, tags and friendship are used in predicting the missing values of user-item matrix efficiently. Empirical analysis on the dataset proves that the proposed approach is better than the existing methods.

**Keywords** Social recommendation • Tags • Personal interest  
Interpersonal influence • Matrix factorization

## 1 Introduction

Recommender systems are one of the most important Web applications that provide many services and suggest some services automatically as per user's interest. Recommender systems have gained its popularity due to information overload prevailing in the Internet. It helps users to identify items which are interested to them. There are various techniques in personalized recommenders like content based and collaborative filtering technique [1]. The traditional recommender systems suffer from various limitations like cold start problem, data sparsity, scalability. To overcome these problems, social recommendation is introduced.

---

K. Vani (✉) · T. Swathiha  
Department of CSE, PSG College of Technology, Coimbatore, India  
e-mail: kvanisaras@gmail.com

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_2](https://doi.org/10.1007/978-981-10-7200-0_2)

Such recommender systems incorporate information like friends, ratings, and tags which improve the accuracy of recommendation. Similarly, users' trust relationship can also be incorporated to improve accuracy of traditional recommender systems. Such recommender systems consider only trustable users and recommend the items rated by them. But it suffers from several limitations. Trust relationships need not be mutual like social relationships, and building such trust network is difficult since very few Websites provide trust assigning facility. Due to the rapid growth of social networking sites, users like to interact more with their friends rather than the trusted users. So these problems led to focus more on social recommender systems. In social recommender systems, not all friends have similar taste as that of the user and different group of friends can contribute to different recommendation tasks. So this paper proposes an approach to group users based on similar items and incorporates efficient regularization [2] to handle friends with different taste. Preference of any item by a user can be known with their ratings as well as from their tagging behaviour [3]. Ratings tell how much a user liked an item, whereas tags indicate why they liked the item [4]. So tagging information is also incorporated with friendship and rating information to make recommendations. Experiments have been conducted on dataset to evaluate the performance of this approach. The paper is organized as follows. Section 2 provides an overview of different approaches of recommender systems. Section 3 describes the social recommendation framework and Sect. 4 presents the experimental analysis and results.

## 2 Related Work

This section reviews important approaches of recommender systems including social recommender systems and tag-based recommender systems.

### 2.1 *Tag-Based Recommender Systems*

Collaborative filtering technique fails when there are diverse domains because people with similar taste in one domain may not agree well in other domain. So to improve recommendation quality, content information about items has been used as additional knowledge which led to the advent of collaborative tagging systems. Tagging information helps in easy retrieval of items [5–7]. It also helps in classifying their favourite items using the keywords. TF-based recommender system is one of the content-based filtering approaches which exploit tagging information. It models tag-based user profiles and item profiles. The simplest method to assign weight to a particular tag is by counting the number of times that tag has been used by the user, or the number of times the item has been annotated by that tag. Items which are highly tagged by the user will be recommended. Probabilistic latent semantic analysis (PLSA) and FolkRank algorithm are used to form tag-based

recommender systems. PLSA [8] is used to find the semantic structure of a data using co-occurrence matrix in a probabilistic framework. FolkRank algorithm [9] is used to perform topic-specific ranking in folksonomies.

## 2.2 Social Recommender Systems

Nowadays, the usage of social networking sites has been increased tremendously. Social influences play a key role when people are making decisions of adopting products. Social recommender systems [10] aim to reduce information overload problem by presenting the most relevant content. MF in social networks is proposed in [11]. It assumes that neighbours in the social network may have similar interest as that of a user. The objective function is

$$\frac{1}{2} \sum_{(i,u)_{observed}} (R_{u,i} - \hat{R}_{u,i})^2 + \frac{\beta}{2} \sum_{allu} \left( (Q_u - \sum_v S_{u,v}^* Q_v) (Q_u - \sum_v S_{u,v}^* Q_v)^T \right) + \frac{\lambda}{2} \left( \|P\|_F^2 + \|Q\|_F^2 \right) \quad (2.1)$$

The second term in the objective function minimizes the distance between user profile  $Q_u$  and average of his friends' profiles  $Q_v$ . Matrix  $S$  denotes the user-user trust values. This equation can be optimized using gradient descent approach. CircleCon model [12] is found to be better than matrix factorization methods [13]. It is an extension of MF in social networks with inferred circle of friends. MF in social networks considers all friends, but preference of friends will be different in different categories. This idea is incorporated in CircleCon model. It infers a circle of friends from entire social network regarding a specific category of items.

Most of the approaches consider only the social network information. Social context factors like individual preference and interpersonal influence are ignored. But these factors which affect user's decision are incorporated in ContextMF model [14]. In ContextMF recommendation, a user sends or recommends a set of items to another user. Percentage of items adopted by user  $U$  from user  $V$  is denoted by interpersonal influence. Matrix  $W$  denotes the interpersonal similarity of users. All these factors contribute in predicting the missing values in user-item rating matrix. Individual preference is obtained by the user's history. The objective function of ContextMF model is

$$\frac{1}{2} \sum_{(u,i)} (R_{u,i} - \hat{R}_{u,i})^2 + \frac{\beta}{2} \sum_u \left( \left( Q_u - \sum_v S_{u,v}^* Q_v \right) \left( Q_u - \sum_v S_{u,v}^* Q_v \right)^T \right) + \frac{\gamma}{2} \sum_{u,v} (W_{u,v}^* - Q_u Q_v^T) + \frac{\lambda}{2} \left( \|P\|_F^2 + \|Q\|_F^2 \right) \quad (2.2)$$

The third term in the Eq. (2.2) denotes the individual preference. Individual preference in ContextMF model is highly related with similarity with other users rather than his own interest. So another personalized recommendation approach [15] is proposed which fuses three social factors, personal interest of a user, interpersonal interest similarity and interpersonal influence. Previous works considered friends or inferred circle of friends for recommendation which solves cold start problem. But it affects the individuality of experienced users. So for experienced users, personal interest of them is given more weightage, and for cold users, preference of friends is given more weightage and considered for recommendation.

### 3 The Recommendation Framework

Users usually turn towards their friends who have similar taste as that of them or who are expert in that field to get suggestions. Users ask suggestions from different set of people for different set of products. So, different group of users contribute to different recommendation tasks. To obtain better results, suitable group of users are clustered and friends of a user within his group are considered to be more similar to the user. Thus, it handles friends with different taste by incorporating it as a regularization term in matrix factorization. Items which are highly rated or tagged by the friends of a user are recommended to him in a social recommender system. This achieves better results for cold users because they don't have any history of rating records but will have adverse effects for existing users. Existing users who have rated comparatively higher number of items will have their own individual preference. In social recommender systems, such individual preference is ignored. So this individual preference of existing users is also retained by incorporating it as another regularization term.

#### 3.1 The Clustering

Figure 1 shows the detailed flowchart of the proposed approach. Items are clustered into stable groups based on the tags. If two items are assigned with same set of tags, then they are considered to be similar. Let  $I_g = \{I_{g_1}, I_{g_2}, \dots, I_{g_p}\}$  denote the set of item groups. Then the tag frequency of every user to the items in every group is calculated.

$$Tag\ frequency_{u, g_i} = \frac{N_{u, g_i}}{N_u} \quad (3.1)$$

where  $Tag\ frequency_{u, g_i}$  represents the frequency of tags given by user  $u$  to items in group  $g_i$  and  $N_u$  denotes the total number of times user  $u$  has tagged items and  $N_{u, g_i}$  denotes the number of times user  $u$  has tagged items in item group  $g_i$ . Users are

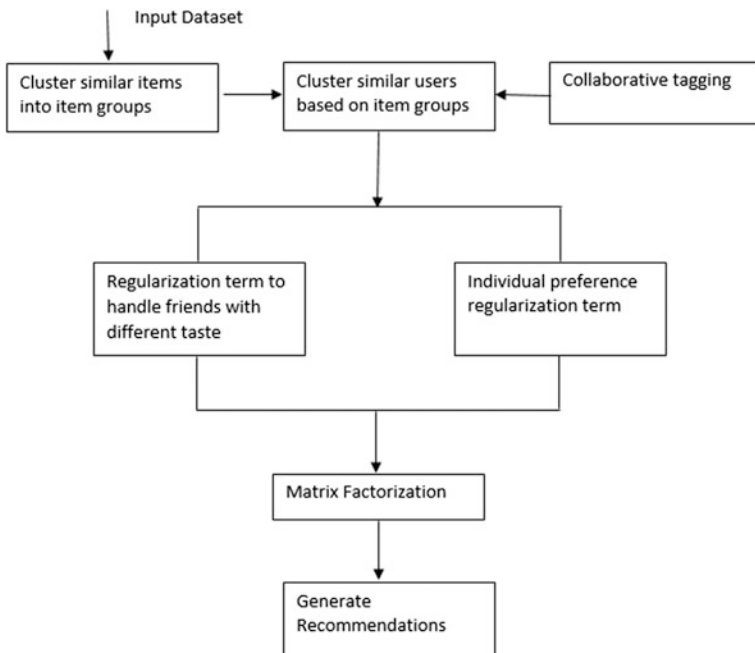


Fig. 1 Flowchart of the proposed approach

related to different item groups with different weightage of tag frequency. Users with similar tag frequencies are clustered into user groups.

### 3.2 Social Regularization

In this approach, friends of a user within the same user group as that of the user will be more similar to him. So friends within the same group are given higher weightage than friends in other groups. The interpersonal interest similarity can be calculated as follows.

$$S(i, f) = \beta \sum_{f \in F_g + (i)} \cos(U_i, U_f) + (1 - \beta) \sum_{f \in F_g - (i)} \cos(U_i, U_f) \quad (3.2)$$

The social regularization term [2] is

$$\frac{\beta}{2} \sum_{i=1}^m \sum_{f \in F(i)} S(i, f) \|U_i - U_f\| \quad (3.3)$$



where  $\beta$  is the weightage given to friends within the same group and  $(1-\beta)$  is the weightage given to friends in other user groups.  $F_g + (i)$  is the set of friends of user  $i$  within the same group of user  $i$  and  $F_g - (i)$  is the set of friends of user  $i$  in other groups.  $S(i, f)$  denotes the cosine similarity between user  $i$  and his friend user  $f$ . If  $S(i, f)$  is greater, then the friend  $f$  contributes more towards the preference of the user.

### 3.3 Individual Preference

Users with higher rating history will usually tend to choose items by themselves with little influence by their friends. The proposed approach retains the individual preference of experienced users with little influence from friends and considers taste of their friends for cold users. Personal interest of user is represented as

$$Q_{u,i}^g = Sim(U_u, V_i) \quad (3.4)$$

It denotes the similarity between feature of user and items in the group  $g$  to which the user is more related [15]. The individual preference regularization term is

$$\frac{\eta}{2} \sum_{u,i} |H_u^g| (Q_{u,i}^g - U_u V_i^T)^2 \quad (3.5)$$

where  $|H_u^g|$  is the number of items the user tagged in the group  $g$ . It denotes how much the user depends on his own preference. Higher the  $|H_u^g|$  value, higher is the impact of individual preference.

### 3.4 Model Training

The objective function of the proposed recommendation model is

$$\begin{aligned} \mathcal{M} = \min_{U, V} & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (R_{ij} - U_i^T V_j)^2 + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \\ & + \sum_{i=1}^m \frac{\beta}{2} \sum_{f \in F(i)} S(i, f) \|U_i - U_f\| + \frac{\eta}{2} \sum_{u,i} |H_u^g| (Q_{u,i}^g - U_u V_i^T)^2 \end{aligned} \quad (3.6)$$

It incorporates both social regularization and individual preference terms mentioned above. The objective function is minimized by gradient descent approach. The gradients of the objective function with respect to  $U_u$  and  $V_i$  are

$$\frac{\partial \mathcal{M}}{\partial U} = \sum_{j=1}^n (U_i^T V_j - R_{ij}) V_j + \lambda U_i + \beta \sum_{f \in F(i)} S(i, f) (U_i - U_f) + \eta \sum_i |H_u^g| (U_u V_i^T - Q_{u,i}^g) V_i \quad (3.7)$$

$$\frac{\partial \mathcal{M}}{\partial V} = \sum_u (U_i^T V_j - R_{ij}) U_i + \lambda V_j + \eta \sum_i |H_u^g| (U_u V_i^T - Q_{u,i}^g) U_u \quad (3.8)$$

The value of user and item latent feature matrices are updated with the gradient values mentioned in the Eqs. (3.7) and (3.8).

## 4 Experimental Results

In this section, experiments are conducted on Last.fm dataset [16] to evaluate the proposed approach and it is implemented in RStudio. Last.fm is a social music Website. It has 1892 users, 17632 unique artists, 11946 tags and 186479 tag assignments. It is the only dataset which has both tags and social contacts. So it is used for this study.

### 4.1 Metrics

Metrics like root mean square error (RMSE), mean absolute error (MAE), precision and recall are used. Precision and recall are defined as,

$$Precision = \frac{|Relevant\ items\ retrieved|}{|retrieved\ items|} \quad (4.1)$$

$$Recall = \frac{|Relevant\ items\ retrived|}{|relevant\ items|} \quad (4.2)$$

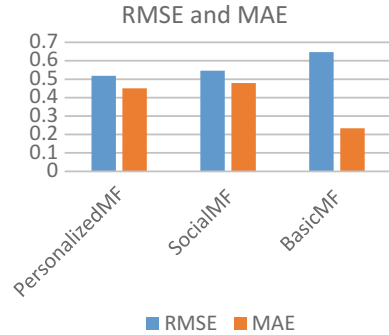
### 4.2 Comparisons

PersonalizedMF is the proposed approach which is compared with BasicMF, SocialMF, TF-based recommender system. BasicMF is the one which employs basic matrix factorization, and SocialMF employs matrix factorization with only social regularization term, whereas the PersonalizedMF incorporates both the social regularization term and individual preference term. TF-based system recommends items which are highly tagged by the users. Learning rate is set to 0.015,

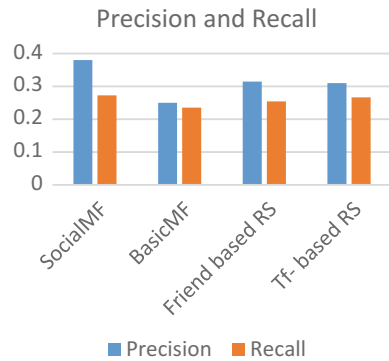
**Table 1** Performance of PersonalizedMF in different  $\beta$  and  $\eta$  values

Weightage	0.5	0.6	0.7	0.8	0.9
RMSE	0.5184	0.5181	0.5179	0.5238	0.5239
MAE	0.4512	0.4509	0.4508	0.4545	0.4545

**Fig. 2** Comparison of different models with RMSE and MAE values



**Fig. 3** Comparison of different models with precision and recall values



and overfitting regularization term  $\lambda$  is set to 0.001. Experiments are conducted by assigning same weightage to both social regularization and individual preference regularization terms. Table 1 depicts the performance of PersonalizedMF in different weightage ( $\beta$  and  $\eta$ ) values. Error is low when weightage is 0.7. Figures 2 and 3 depict the performance comparison of various methods with the metrics.

## 5 Conclusion and Future Work

In this paper, we focus on improving the quality of social recommender system. The existing social recommendation approaches use social relationship of user without considering the interpersonal similarity. But some social connections may have adverse effects in the recommendation. In reality, every user will have different

circle of people like family, friends, colleagues, neighbours and so on. User will consult different group of people to get suggestions for different category of items. So the proposed approach clusters items and similar users based on their tag frequency to find different group of friends for different recommendation tasks and incorporates social regularization term to handle friends with different taste and individual preference term to retain the preference of existing users. The experiments on the benchmark dataset show that the proposed approach is better than the existing approaches. Time and location information of the user can also be considered for recommendation.

## References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
2. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: *Proceedings of the 4th ACM International Conference on Web Search Data Mining*, Hong Kong, China (2011)
3. Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: *SAC'08: Proceedings of the 2008 ACM Symposium on Applied Computing* (2008)
4. Milicevic, A.K., Nanopoulos, A., Ivanovic, M.: Social tagging in recommender systems: a survey of the state of art and possible extensions. *Artif. Intell. Rev.* **33**(3) (2010)
5. Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social media recommendation based on people and tags. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 194–201 (2010)
6. Liang, H., Xu, Y., Li, Y., Nayak, R.: Collaborative filtering recommender systems using tag information. In: *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 59–62 (2008)
7. Chatti, M.A., Dakova, S., Thus, H., Schroeder, U.: Tag based collaborative filtering recommendation in personal learning environments. *IEEE Trans. Learn. Technol.* **6**(4) (2013)
8. Hofmann, T.: Probabilistic latent semantic analysis. In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 31 July–06 Aug, pp. 688–693 (1999)
9. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: FolkRank: a ranking algorithm for folksonomies. In: *Proceedings of Workshop on Information Retrieval (FGIR)*, Germany (2006)
10. Liu, F., Lee, H.J.: Use of social network information to enhance collaborative filtering performance. *Expert Syst. Appl.* 4772–4778 (2010)
11. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: *Proceedings of the 4th ACM Conference on Recommender Systems*, Barcelona, Spain, pp. 135–142 (2010)
12. Yang, X., Steck, H., Liu, Y.: Circle-based recommendation in online social networks. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp. 1267–1275, Aug 2012
13. Koren, Y., Yahoo Research, Bell, R., Volinsky, C., AT&T Labs Research.: Matrix factorization techniques for recommender systems. *Computer* **42**(8) (2009)

14. Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., Yang, S.: Social contextual recommendation. In: Proceedings of the 21st ACM International Conference on Information and knowledge Management, Maui, Hawaii, USA (2012)
15. Qian, X., Feng, H., Zhao, G., Mei, T.: Personalized recommendation combining user interest and social circle. *IEEE Trans. Knowl. Data Eng.* **26**(7) (2014)
16. <http://grouplens.org/datasets/hetrec-2011/>

# Twitter Sentimental Analysis on Fan Engagement



Rasika Shreedhar Bhangle and K. Sornalakshmi

**Abstract** Sentimental analysis involves determination of opinions, feelings, and subjectivity of text. Twitter is a social networking service where millions of people share their thoughts. Twitter sentimental analysis on fan engagement focuses on how fans in different sports industry actively engage on social media. Fans are identified based on their opinions and emotions expressed in the tweet. In this paper, we provide a comparative analysis of machine learning algorithms such as multinomialNB, support vector machine, linearSVM, and decision trees. The traditional approach involves manually assigning labels to training data which is time-consuming. To overcome this problem, we use TextBlob which computes the sentiment polarity based on POS tagging and assigns sentiment score to every tweet in range of  $-1$  to  $+1$ . We compute the subjectivity of text which is in the range of  $1-0$ . We also identify the sarcastic tweets and assign correct class labels based on the weightage we provide for every word. Our experimental result shows how accuracy gets increased with the use of advanced machine learning algorithms.

**Keywords** Twitter • Sentimental analysis • Machine learning • Analytics  
Naïve bayes • MultinomialNB • Support vector machine • LinearSVM  
Decision trees • Polarity analysis • MongoDB • Python

## 1 Introduction

Today, the major professional sports have their analytics department and experts on staff. The popularity of data-driven decision making in sports has increased the general awareness about fans and their likings, which are consuming more ana-

---

R. S. Bhangle (✉)

Department of Information Technology (Big Data Analytics), SRM University,  
Kattankulathur, Chennai 603203, Tamil Nadu, India  
e-mail: rasikabhangle@gmail.com

K. Sornalakshmi

Department of Information Technology, SRM University, Kattankulathur,  
Chennai 603203, Tamil Nadu, India

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_3](https://doi.org/10.1007/978-981-10-7200-0_3)

lytical content than ever. Fans act as the primary resource when one is looking to capitalize on the business of sports. Moreover, fans are interactive in nature, and therefore, they can influence at any level of business model in sports. Fan data is an aggregation of their feelings and expectations of the team's performance and achievements. This data could come in from channels like Twitter, Tumblr, Instagram, and Facebook. Teams consume this data in order to make quick clear decisions that resonate with the fans' feelings.

We can use fan data consumption for customer segmentation. With big data, we can not only track what kind of data fans are accessing but also have a location and time-based segmentation. We can cluster fans in different buckets and produce the content based on the likes and dislikes of that bucket. Upon clustering, we can have an overview of the kind of products and services the fans consume. This can help in developing similar content to increase engagement level.

In this paper, we look at one such popular blog called as Twitter. Our focus is to identify the fans by gathering all the tweets and assigning a category to each. The categories are positive, negative, and neutral. This shows the engagement of individuals and allows us to track the actively participating fans. We are also trying to identify the most trending topics used by fans.

## ***1.1 Introduction to Sentimental Analysis***

Sentimental analysis is a process of collecting and analyzing data based on the person's feelings, reviews, opinions, and thoughts. Sentimental analysis can be done at document, phrase, and sentence level. We have performed sentence-level sentiment classification on tweets which involves sentiment extraction, sentiment classification, sentiment score, subjectivity classification, summarization of opinions, and most informative features.

## ***1.2 Introduction to Sklearn***

Scikit-learn (sklearn) is an efficient tool for data mining and analysis. It is an open-source Python library that implements a wide range of machine learning techniques. We have used sklearn with Python 3.5 version. It is built on numpy, scipy, and matplotlib. We have used sklearn for preprocessing, model selection, classification, cross-validation, and visualization of tweets. We have worked on various classification algorithms including multinomialNB, support vector machines, linearSVM, and decision trees.

## 2 Literature Survey

Many researches have been done on sentimental analysis in the past. Nowadays, millions of messages are generated by users from many social networking Web sites like Facebook, Twitter, Amazon, and Google Plus. Pang [1], Lee, and Vaityanathan were the first to work on sentimental analysis. Their main aim was to classify text by overall sentiment and not just by topic. They applied machine learning algorithm on movie review database which resulted that these algorithms outperform human-produced algorithms.

Loper and Bird [2] proposed Natural Language Toolkit (NLTK) which is a library that consists of many program modules, large set of structured files, problem sets, statistics functions, machine learning classifiers, etc. The main purpose of NLTK is to carry out natural language processing, i.e., to perform analysis on human language data.

Wang et al. [3] were the researchers who proposed a system for real-time analysis of public responses for 2012 presidential elections in USA. They collected the responses from Twitter. Twitter is one the social network sites where people share their views, thoughts, and opinions on any trending topic. A relation was created between sentiments that arose from public response on Twitter with the complete election events.

Almatrafi et al. [4] were the researchers who proposed a system based on location. According to them, sentimental analysis is carried out by natural language processing (NLP) and machine learning algorithms are used to extract a sentiment from a text. They studied various applications of location-based sentimental analysis by using a data source in which data can be extracted from different locations easily.

Jiang et al. [5] created a system which focused on target-dependent classification. It was based on Twitter in which a query is given first and then classified as positive, negative, or neutral sentiments with respect to that query that contains sentiment. In their research, query sentiment served as target. The target-independent strategy was adopted to solve these problems with the help of state-of-the-art approaches, which may sometime assign immaterial sentiments to target.

Tan et al. [6] worked on sentimental analysis, and their research stated that the information can be used to improve user-level sentimental analysis. Their base of research was social relationships. They used Twitter as their source of experimental data and used semi-supervised machine learning framework to carry out analysis.

Pak and Paroubek [7] performed linguistic inspection of collected corpus and built a sentiment classifier that was used to determine positive, negative, and neutral sentiments of Twitter document. They also proposed a system for emoticons used in tweets, in which they created a corpus for emoticons such that they replaced each emoticon with their respective meaning so that it can extract feature from emoticons.



Minging and Bing [8] tried to show how sentimental analysis influences from social network posts and also to compare the result on various topics on different social-media platforms.

### 3 Methodology

#### 3.1 Data Description

For analysis purpose, we have collected two data sets. We extracted Twitter data using Firehose API and stored it into MongoDB. We have performed sentimental analysis using Spyder IDE which is a Scientific Python Development Environment and is included with Anaconda. Anaconda is an open data science platform powered by Python 3.5 (Table 1).

#### 3.2 System Workflow

Figure 1.

#### 3.3 Text Mining and Preprocessing

A tweet contains lots of opinions and views which are expressed by different users. These tweets are preprocessed primarily to prepare them for classification. Text preprocessing includes the following:

- Converting Twitter text to normal mode with decode method
- Escape html entities
- Remove all URLs including http links and hash tags
- Remove stop words and separators
- Remove punctuations and multiple white spaces
- Eliminate tweets that are not in English
- Replace all the emoji with their sentiments.

**Table 1** Example data set

Data sets	Number of tweets
Cricket	8,000
National basketball association (NBA)	25,000
Formula one (F1)	96,406

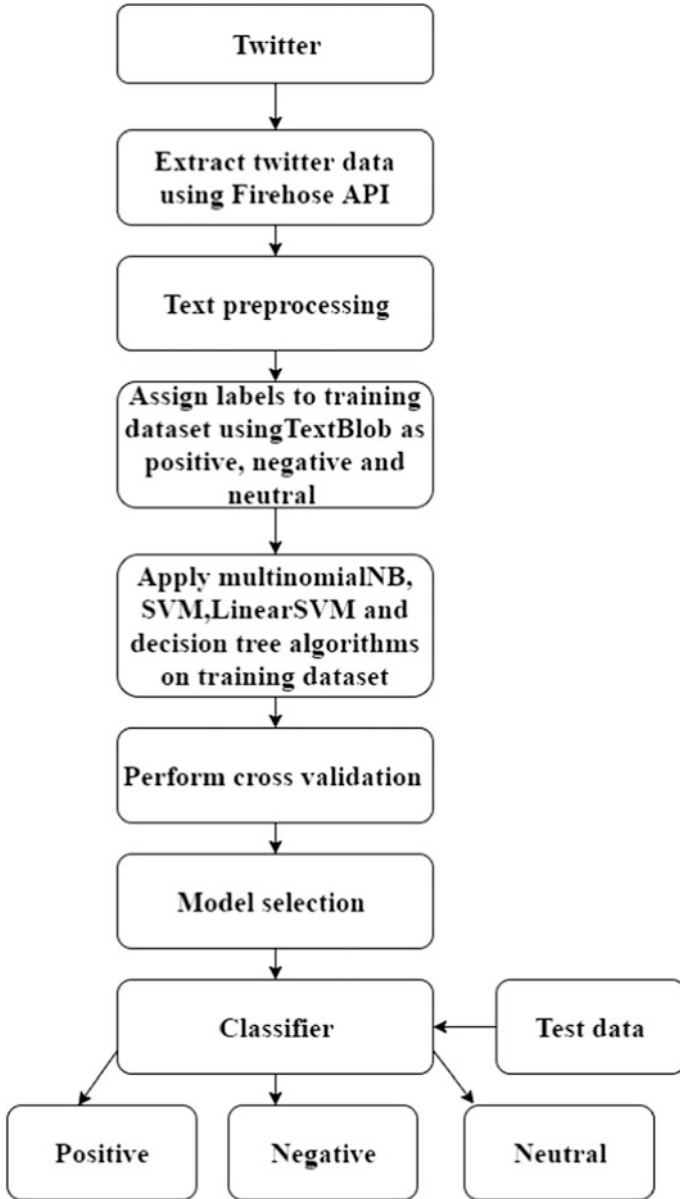


Fig. 1 Sentimental analysis architecture

### 3.4 *Feature Selection*

- **Parts Of Speech Tagging**  
The function of a word in a sentence is called as parts of speech. It tells which word functions as a noun, pronoun, preposition, verb, determiner, adjective, and any other parts of speech. POS tags are good indicators of subjectivity. We use POS tags as features because the same word may have many different meanings depending on its usage.
- **Term presence and their frequencies**  
These features are individual words or n-gram words with their frequency counts. It uses the term frequency weights to indicate relative importance of features.
- **Negation**  
The appearance of negative word usually changes the polarity of the opinion. We have tried to handle negations for our classification.

### 3.5 *Sentiment Polarity*

After all the preprocessing tasks and feature selection, each tweet is processed to calculate the polarity of the sentence. This is achieved using TextBlob, a Python library for processing textual data. It tells us what part of speech each word in a text corresponds to. It has a sentiment property which returns a named tuple of form sentiment (polarity and subjectivity). The polarity score is a float within the range  $[-1.0, 1.0]$ . The subjectivity is a float within the range  $[0.0, 1.0]$  where 0.0 is very objective and 1.0 is very subjective. With the help of TextBlob, we also try to attempt spelling correction. The spelling correction is based on Peter Norvig's concept and is about 70% accurate.

## 4 **Training**

We have trained the classifiers to discover predictive relationships for unknown data. During training phase, only the training set is available which is different from test data set. The test set will be available only during testing the classifier. We have followed 80–20 rule where 80% of data is kept for training the classifier and rest 20% of data for testing the accuracy of model.

## 5 Sentiment Classification

### 5.1 *Naïve Bayes Classifier*

We first built a sentiment classifier using simple Naïve Bayes. Naïve Bayes classifiers are probabilistic classifiers with strong independence assumptions between features. It is a simple technique for constructing classifiers. With simple Naïve Bayes, we got less accuracy as the tweets were classified into wrong labels. To improve more on accuracy, we used Multinomial Naïve Bayes.

### 5.2 *Multinomial Naïve Bayes Classifier*

Multinomial Naive Bayes is a specialized version of Naive Bayes that is designed for text. Simple Naive Bayes would model a document as the presence and absence of particular words, whereas Multinomial Naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with. It implements Naïve Bayes for data distributed multinomially and also uses one of its version for text classification in which word counts are used to represent data. With MultinomialNB classifier, we could achieve higher accuracy.

### 5.3 *Support Vector Machine*

Support vector machine is a supervised machine learning algorithm for classification. With SVM, we plot each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. We have performed classification by finding the hyper-plane that differentiates the three classes. Using sklearn, we applied SVM classifier to our data set. Simple SVM implementation is based on libsvm. The multiclass support is handled according to a one-vs-one scheme. With support vector classification (SVC), we observed that the accuracy gets decreased and generates incorrect class labels.

### 5.4 *Linear Support Vector Machine*

Linear support vector machine is similar to SVC with parameter kernel = linear, but is implemented in terms of liblinear rather than libsvm. So it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. This class supports both dense and sparse input, and the

multiclass support is handled according to a one-vs-the-rest scheme. With LinearSVM, we got the best accuracy for our classification problem and most correct class labels for our test data set.

## 5.5 Decision Tree Classifier

Decision trees are nonparametric supervised learning method used for classification. The goal of it is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Decision tree classifier is capable of performing multiclass classification on a data set. With DT, we are getting 100% accuracy but incorrect class labels for our test data set. This indicates overfitting of model.

## 6 Classification Evaluation

The performance of sentiment classification is calculated by:

$$\begin{aligned} \text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ \text{Precision} &= \text{TP} / (\text{TP} + \text{FP}) \\ \text{Recall} &= \text{TP} / (\text{TP} + \text{FN}) \\ \text{F1} &= (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \end{aligned}$$

## 7 Experimental Results

A collection of texts is also called as corpus. The two data sets tennis and cricket have been labeled using TextBlob. This is our labeled training set. We have trained machine learning model to learn to discriminate between positive/negative/neutral automatically. Then, with a trained model, we are able to classify the unlabeled messages as either positive or negative or neutral. We then used the bag-of-words approach, where each unique word in a text is represented by a numeric value. We then normalized the words into their base form. Next, we converted each message, represented as a list of tokens above into a vector that machine learning models can understand. This involves counting the occurrences of each word in message, assigning weights to counts, and normalizing the vectors to unit length. (Table 2).

**Table 2** Confusion Matrix

	Predicted: NO	Predicted: YES
Actual: NO	True negatives	False positives
Actual: YES	False negatives	True positives

Here, we have used scikit-learn (sklearn), a powerful Python library for teaching machine learning. After the counting, the term weighting and normalization is done with TF-IDF, using scikit-learn’s TfidfTransformer. We then applied classification algorithms and compared the results. The result gives the predicted class labels for every tweet.

Initially, we started our analysis with 8,000 tweets and then increased our data set to 96,406 tweets. We performed classification without cross-validation and checked for the model accuracy (Tables 3 and 4).

The result below shows the three-class classification of F1 data set (Fig. 2 and Table 5).

According to our experimental survey, LinearSVM works best, thereby giving the most accurate class labels for each tweet (Fig. 3).

From this confusion matrix, we compute precision and recall and their combination f1-score which is harmonic mean (Tables 6 and 7).

We perform cross-validation of our SVM classification model to avoid overfitting. For this, we split the data set into training (80%) and testing (20%). We compute the term frequency, inverse document frequency, and normalizing vectors and passed them into a single pipeline. Then, we performed tuning of hyper-parameters gamma and C of RBF of kernel SVM for correcting misclassification. The C and gamma functions are for nonlinear support of SVM. Next, we performed k-fold cross-validation where the value of k is either 3, 5, 7, or 10 for model stability. For each of the k-folds, a model is trained using k-1 of the folds as training data. The resulting model is validated on the remaining part of the data. These estimators have a score method which can judge the quality of the fit. We then fit an estimator API called as grid searchCV on all possible combinations of parameter values and retain the best possible combination for model optimization.

We then fit the grid svm model on our tweets and corresponding labels. The resulting predictor is serialized to disk which creates an image file of the trained data. So next time we can reload the trained model directly. With iterative model tuning and parameter search, we are able to get an optimized classification model (Fig. 4 and Table 8).

**Table 3** Accuracy calculation of cricket data set

Algorithm	Data set	Accuracy (%)
MultinomialNB	8,000	85
SVM	8,000	47
LinearSVM	8,000	96
Decision tree	8,000	100

**Table 4** Accuracy calculation of NBA data set

Algorithm	Data set	Accuracy
MultinomialNB	25,000	87
SVM	25,000	53
LinearSVM	25,000	97
Decision tree	25,000	100

**Fig. 2** Statistical aggregation of F1 data set

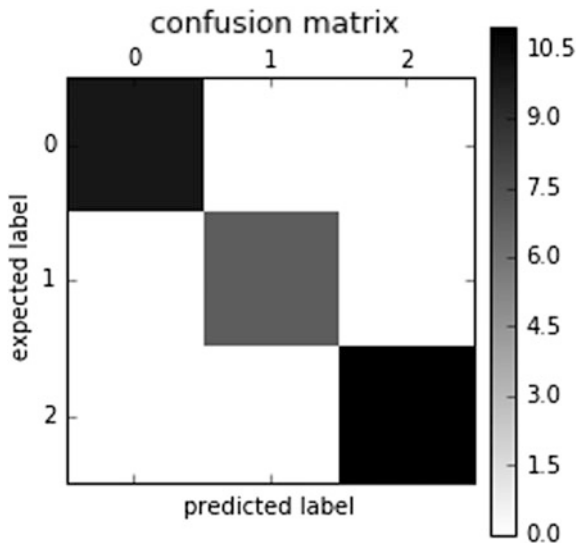
```

Data discription
message
label
negative count      6022
              unique  4683
              top      is worse
              freq     129
neutral count      8181
              unique  5445
              top      japanesegp
              freq     114
positive count     82203
              unique  4410
              top      is outstanding
              freq     759
    
```

**Table 5** Accuracy calculation of F1 data set

Algorithm	Data set	Accuracy
MultinomialNB	96,406	89
SVM	96,406	58
LinearSVM	96,406	99
Decision tree	96,406	100

**Fig. 3** Confusion matrix for three-class classification



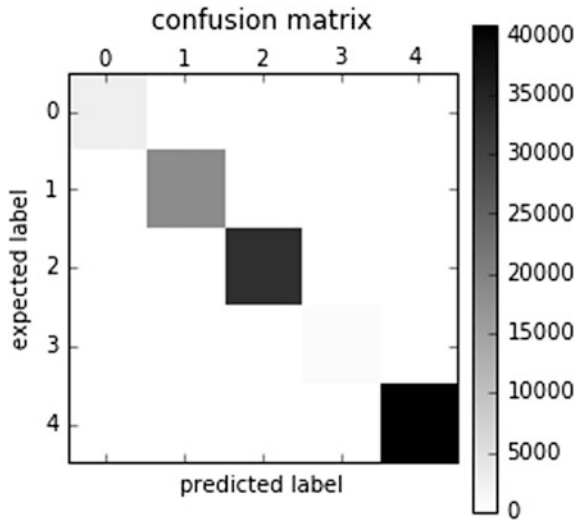
**Table 6** Three-class classification report of F1 data set

	Precision	Recall	f1-score	Support
Negative	0.99	0.98	0.98	6022
Neutral	0.97	0.97	0.97	8181
Positive	1.00	1.00	1.00	82203
Total	0.99	0.99	0.99	96406

**Table 7** Confusion matrix of F1 data set

5874	76	72
62	7920	199
24	147	82032

**Fig. 4** Confusion matrix for five-class classification



**Table 8** Five-class classification report of F1 data set

	Precision	Recall	f1-score	Support
Mildly negative	0.99	0.97	0.98	2780
Mildly positive	1.00	0.99	1.00	18589
Neutral	1.00	1.00	1.00	33349
Strongly negative	0.99	0.99	0.99	731
Strongly positive	1.00	1.00	1.00	40957
Total	1.00	1.00	1.00	96406



## 8 Conclusion

We presented results for Twitter sentimental analysis on fan engagement. Based on POS tagging and the weightage assigned for every word, we could analyze the category of tweets. Our classifier is able to determine the positive, negative, neutral, and sarcastic tweets. The classifier is based on linear and nonlinear SVM that uses BOW and POS tags as features. We noticed that the maximum number of tweets was categorized as strongly positive tweets. We also observed that retaining the emoji helped us in understanding the mental activity of every fan.

As the future work, we plan to collect more data and apply deep learning concept to our problem.

## References

1. Pang, P., Lee, L., Vaithyanathan, S.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts (2004)
2. Loper, E., Bird, S.: NLTK library
3. Wang, H., Can, D., Bar, F., Narayana, S.: Sentimental analysis of twitter data using NLTK
4. Almatrafi, O., Parack, S., Chavan, B.: Combining rule based classifiers
5. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent Twitter Sentiment Classification
6. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: Extraction and mining of academic social networks
7. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of LREC (2010)
8. Minging, H., Bing, L.: Mining and summarizing customer reviews. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04) (2004)
9. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp. 1320–1326 (2010)
10. Go, A., Bhayani R., Huang L.: Twitter sentiment classification using distant supervision. Technical Paper, Stanford University (2009)
11. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In: Proceedings of the 13th International Conference on Discovery Science, pp. 1–15. Springer, Berlin, Germany (2010)
12. Neethu, M.S., Rajashree, R.: Sentiment analysis in twitter using machine learning techniques. In: 4th ICCNT. Tiruchengode, India. IEEE—31661 (2013)
13. Peddinti, V.M.K., Chintalapoodi, P.: Domain adaptation in sentiment analysis of twitter. In: Analyzing Microtext Workshop, AAAI (2011)
14. Dumais, S. et al.: Inductive learning algorithms and representations for text categorization. In: Proceedings of the Seventh International Conference on Information and Knowledge Management. ACM (1998)
15. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technical Report, Stanford (2009)
16. Wilson, T., Wiebe, J., Hoffman, P.: Recognizing contextual polarity in phrase level sentiment analysis. ACL (2005)

17. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press (1999)
18. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the Conference on Web Search and Web Data Mining (WSDM) (2008)
19. Popescu, A.-M., Etzioni, O.: Extracting Product Features and Opinions from Reviews. EMNLP-05 (2005)
20. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of International Conference on Machine Learning (ICML'01) (2001)
21. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical co-occurrence. *Behav. Res. Methods* **28**, 203–238 (1996)
22. Zhou, L., Li, B., Gao, W., Wei, Z., Wong, K.: Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: Presented at the 2001 Conference on Empirical

# A Hybrid Semantic Algorithm for Web Image Retrieval Incorporating Ontology Classification and User-Driven Query Expansion



Gerard Deepak and J. Sheeba Priyadarshini

**Abstract** There is always a need to increase the overall relevance of results in Web search systems. Most existing web search systems are query-driven and give the least preferences to the users' needs. Specifically, mining images from the Web are a highly cumbersome task as there are so many homonyms and canonically synonymous terms. An ideal Web image recommendation system must understand the needs of the user. A system that facilitates modeling of homonymous and synonymous ontologies that understands the users' need for images is proposed. A Hybrid Semantic Algorithm that computes the semantic similarity using APMI is proposed. The system also classifies the ontologies using SVM and facilitates a homonym lookup directory for classifying the semantically related homonymous ontologies. The users' intentions are dynamically captured by presenting images based on the initial OntoPath and recording the user click. Strategic expansion of OntoPath based on the user's choice increases the recommendation relevance. An overall accuracy of 95.09% is achieved by the proposed system.

**Keywords** Homonyms • Image retrieval • Ontologies • Recommendation systems • SVM • Web image mining

## 1 Introduction

Web search is an important technique for retrieving the required information from the World Wide Web. The Web is an affluent informational repository [1] where the information density is the highest. Web searches take place through search engines [2] which are an implementation of several Web mining techniques and algorithms

---

G. Deepak (✉)

Department of Computer Science and Engineering, Faculty of Engineering,  
Christ University, Bangalore, India  
e-mail: gerry.deepu@gmail.com

J. Sheeba Priyadarshini

Department of Computer Science, St. Josephs College, Bangalore, India

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_4](https://doi.org/10.1007/978-981-10-7200-0_4)

that makes the mining of the Web quite easy and efficient. When a Web page or a Web content document need to be retrieved, a simple approach involving calculating the semantic similarity [3] between the query and Web documents, probabilistic matching [4], keyword matching [5], etc., is traditionally applied in order to retrieve the text content, whereas when an image relevant to a query has to be retrieved a problem arises. There are several terms which are homonyms which may be spelt the same but have different meaning. Image search is an important application of Web mining where the search engine must be able to extract the required images as per the query in a manner such that the images obtained must be highly relevant to the query that is input by the user. The user's intention as well as the retrieved images must have a high degree of correlation. Also, the search engine must be able to distinctly retrieve all the unique images for the query involving homonyms, synonyms, and also, several unique elements for the search query must be displayed. The users' intention must act as a driving force to display the images of high correctness and satisfy the users' need for image search. The search efficiency must still be maintained, i.e., the number of relevant images for a specific query based search must be maximized.

**Motivation:** Most of the existing systems for Web image retrieval have a query-oriented perspective [6] and are not user-oriented. The ultimate goal of any retrieval system must be based and directed as per the users' choice, thus satisfying the user's need for the images. Certain existing systems which capture user's preferences still do not make a mark as they neglect the perspective of the user to give the best results. This enhances the noise [7] of Web image search and increases the irrelevance of images retrieved in the context of Web image search that needs to be overcome.

**Contribution:** An ontological approach that deals with modeling appropriate ontologies for homonyms is proposed. Aggregation of several semantically similar classes and then establish a hierarchical pathway for ontologies for classifying images based on the query input. The proposed system also captures the individual user's choice and then retrieves all the possible images based on the user's intention. The proposed system also incorporates SVM for classifying the ontologies based on their query terms. An APMI strategy is incorporated for semantic similarity computation. Also, the incorporation of homonym lookup table reduces the overall response time of the recommendation process.

**Organization:** This paper is organized as follows. Section 2 provides an overview of the related research work. Section 3 presents the proposed system architecture. Implementation is discussed in the Sect. 4. Performance evaluation and results are discussed in Sect. 5. This paper is concluded in Sect. 6.

## 2 Related Work

Kousalya and Thananmani [8] have put forth content-based image retrieval technique with multifeature extraction that involves the extraction of graphical features from the image. Euclidean distance is used for similarity computation in this approach. Dhonde and Raut [9] used the hierarchical k-means strategy for the retrieval of images from the Web and have increased the overall Web search proficiency. Umaa and Thanushkodi [10] proposed a content-based image retrieval technique using hybrid methodologies. Amalgamation of several methods like cosine transforms, wavelet transforms, feature point extractions, Euclidean distance computations is incorporated. Deepak and Andrade [11] have proposed OntoRec algorithm that incorporates NPMI technique with dynamic Ontology modeling for synonymous ontologies. Deng et al. [12] have proposed the Best Keyword Cover for searching using keywords with minimum inter-object distance. A nearest neighbor approach is imbibed for increasing the overall entities in Web search. The drawback of strictly query-specific Web search is not overcome here without any preference given to the users' choice. Ma et al. [13] have measured ontologies by achieving normalization of ontologies. Ontology normalization refers to removing those ontologies which are not a best fit to a domain. Shashi S et al. have proposed a novel framework using multi-agents for Web image recommendation. An object-centric approach for annotation and crawling of images has been proposed to overcome several existing problems.

Bedi et al. [14] have proposed a focused crawler that uses domain-specific concept type ontologies to compute the semantic relevance. The ontological concepts are further used in expansion of a search topic. Sejal et al. [15] have proposed a framework that recommends images based on relevance feedback and visual features. The historical data of clicked and unclicked images is used for relevance feedback, while the features are computed using the cosine similarity measure. Kalanditis et al. [16] have proposed a paradigm of locally optimized hashing with the justification of the fact that it requires set intersections and summations alone. Also, a clustering-based recommendation has been imbibed into the system using a graph-based strategy. Gerard Deepak and Priyadarshini [17] have proposed an Ontology-driven framework for image tag recommendation by employing techniques like Set Expansion, K-Means Clustering, and Deviation computation. The Modified Normalized Google Distance Measure is employed for computing the semantic deviations. Wang et al. [18] proposed a new methodology of multimodal re-ranking using a graph. This approach encompasses modal weights, distance metrics, and relevance scores together into a single platform. Chu and Tsai [19] have considered visual features for proposing a hybrid recommendation model to predict favorite restaurants. Content-based filtering and collaborative filtering is encompassed together to depict the importance of visual features considered.

### 3 Proposed System Architecture

The proposed system architecture of the Hybrid Semantic Algorithm is depicted in Fig. 1 and comprises of two individual phases. Phase 1 mainly concentrates on building of ontologies for the homonymous search keywords. The Ontology development need not be a homonym always but semantically similar or even slightly related ontological terms with several ontological commitments can be included to produce an essence of similarity searches. The phase 1 implementation is definitely with respect to the ontologies where a semantic meaning is imparted to the Web search algorithm. An ontological strategy is proposed to achieve the possibility of the relevance of images at a single step by mining the possible heterogeneous images based on the Ontological commitments for a specific domain-relevant ontological search term. Several homonyms need to be initially listed and together defined along with the several similar terms and are modeled as ontologies. The homonym lookup directory is a HashMap with a single key but multiple values. The key is the homologous Ontologies and the values are the descriptions of the Ontologies, and this enhances the classification of Ontologies. The various search paths for an individual ontological term is noted, and furthermore, their hierarchy is expanded based on similarity and relevance of search terms.

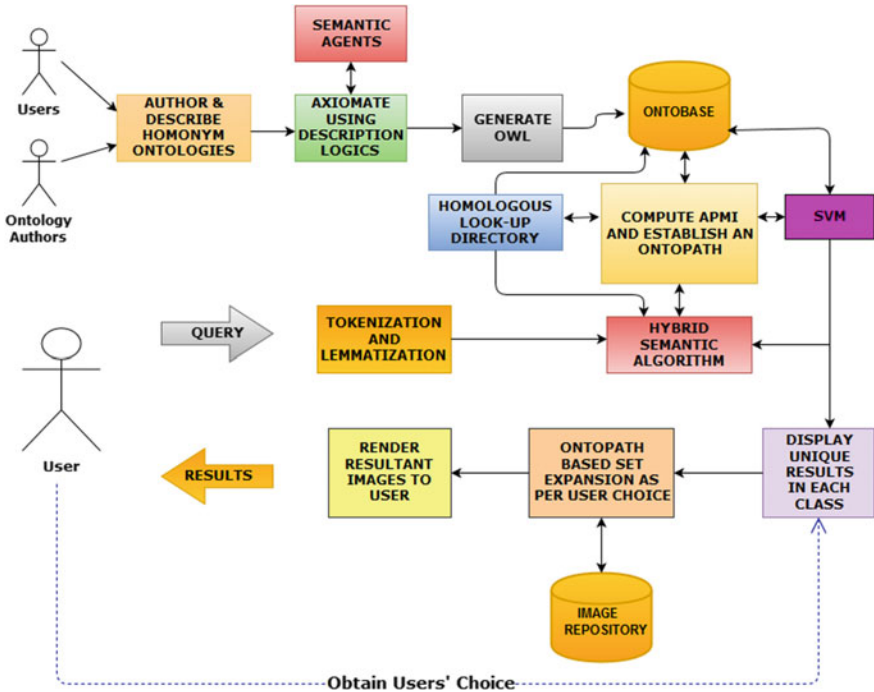


Fig. 1 Proposed architecture of Hybrid Semantic Algorithm

An individual pathway is actually depicted and modeled as OWL ontologies that are used to conceptually establish the OntoPath for the proposed algorithm.

The actual Web image recommendation for user query search takes place in the phase 2 where the system accepts query from the user and processes the query. Upon query preprocessing, the Ontologies in the OntoBase are classified by the hard margin SVM based on the input query words. The semantic similarity is computed between the homologous terms relevant to the query obtained from the lookup directory and the class labels of the Ontologies. Based on the Axiomating Agents and the Description Logics of OWL Semantics, an OntoPath is established by hierarchically arranging the relevant ontologies. Based on the Ontologies in OntoPath, the semantically relevant images are yielded to the user. The user's click on the image is also a driving force for the query to be expanded. The query is then expanded dynamically based on the OntoPath that was formulated recommending the images to the user. There is a dynamic input of user's preferences based on the user click, and thereby a lot of irrelevant images are abstracted from the user increasing the overall recommendation relevance of the system. The semantic similarity is computed using the Adaptive Pointwise Mutual Information (APMI) measure is a modified version of the Pointwise Mutual Information (PMI) measure and is used to compute the semantic similarity. The APMI depicted in Eq. (1) is much better than the other variants of the PMI and is associated with an adaptivity coefficient  $y$ . The adaptivity co-efficient  $y$  depicted in Eq. (2) is associated with a logarithmic quotient in its numerator and its denominator. The adaptive coefficient when coupled with the PMI value enhances the overall performance of the system.

$$APMI(m; n) = \frac{pmi(m; n)}{p(m)(n)} + y \quad (1)$$

$$y = \frac{1 + \log[p(m, n)]}{p(n) \log[p(m)] - p(m) \log[p(n)]} \quad (2)$$

## 4 Implementation

The implementation is accomplished using JAVA as a programming language for the front end. The Ontology definition based on the ontological commitments of the homonymous or synonymous terms is modeled Using Protégé 3.4.8. The rendered Ontologies are in the OWL format, which incorporate the intelligence into the proposed algorithm shown in Table 1. The unstructured image data is stored in the image repository designed using MYSQL. Once the Ontologies are modeled and are integrated within the search environment by automatic Web crawling, the system is ready to query the user preferences for any search term.

**Table 1** Proposed Hybrid Semantic Algorithm for Web page recommendation

<p><b>Input:</b> The query Q which is input as a keyword or a set of multi keyword, Optionally manual ontologies O that includes {O1, O2, O3...} structures for homonyms is included.</p> <p><b>Begin</b></p> <p><b>Step 1:</b> Tokenize and lemmatize the input query Q to obtain a list of query words <math>q_w</math>.</p> <p><b>Step 2 :</b> For each <math>q_w</math>, Look up in Homologous Directory and retrieve the unique classes of Homologous Ontologies from the OntoBase to obtain HashSet <math>H_{gl}</math></p> <p><b>Step 3:</b> Based on <math>q_w</math>. Classify the Ontologies in the OntoBase into <math>C_i</math>; approximate classes using SVM.</p> <p><b>Step 4 :</b> Compute APMI (<math>H_{gl}</math>. elements, <math>C_i</math>. labels)          If <math>APMI &lt; 0.25</math>          HashMap OntoEle(key, value) ← [( <math>H_{gl} / C_i</math> , APMI value)]</p> <p><b>Step 5:</b> Using Axiomatic Agents and OWL Description Logics, Obtain the OntoPath for each OntoEle.</p> <p><b>Step 6 :</b> for each OntoEle          APMI (OntoEle.instances, Image Labels <math>I_L</math>)          If <math>APMI &lt; 0.25</math>          Load the Images to the User</p> <p><b>Step 7:</b> Based on the user click <math>U_c</math> of the various classes of Images Recommended, Expand the Query Based on the established Ontopath Established and Yield the Corresponding Images to the User.</p> <p><b>Step 8:</b> Repeat Step 7 until no further user click.</p> <p><b>End</b></p>
---

## 5 Results and Performance Evaluation

The data sets for the experimentation are collected from the results of Bing and Google Image Search engines. The experimentation was done for 1492 out of which 1321 images were automatically crawled using a customized image crawler, and the remaining images were manually entered into the database. All the images were collected with their labels. Protégé was used for Ontology modeling. The results of various search queries are depicted in Table 2.

The performance is evaluated using precision, recall, and accuracy as metrics for the proposed algorithm and is depicted in Table 3. Standard formulae for precision, recall, and accuracy have been used for evaluating the system performance. The proposed Hybrid Semantic Algorithm yields an average precision of 94.42%, an average recall of 95.76%, and an average accuracy of 95.09%. The reason for a higher performance of the proposed Hybrid Semantic Algorithm is that it uses a homologous lookup directory which reduces the average classification time of the homonymous ontologies. The incorporation of hard margin SVM makes it quite feasible for initial classification of ontologies in the OntoBase. The use of APMI for computing the semantic similarity and capturing of user preferences by dynamic user clicks increases the precision, recall, and accuracy to a larger extent.



**Table 2** Results yielded for various search queries

Search Query	Returned Results
Apple	Apple fruit red, apple fruit green, group of red apples, Apple Laptops, Apple I-Pod, Apple Tablets and I-Pads, Apple I-Phones, Apple fruit Animations, Adams Apple (Image of a Male Throat), a basket of apples, a stall of apples, a cart of apples, apple tree, custard apple, wood apple
Rose	Single red rose, bunch of red roses, bunches and single roses of color yellow, pink, white, orange, blue, hybrid color roses with many colors, roses with dew drops, button roses, rose plant, rose petals, rose water, rose milk, rose wood furniture, wood rose carnation, rose perfumes, images of people with name rose or t-shirts with rose, rose-based sweets
Milk	Glass of milk, carton of milk, bottle of milk, milk products, milk sweets, milk chocolates, milk biscuits, milk brands, images of children drinking milk, milk diary, diary milk chocolates
Net	Fishing net (of different sizes), Internet, network image, images of .NET, netted fabrics, netted attires, mosquito net, UGC NET Advertisement

**Table 3** Performance analysis of Hybrid Semantic Algorithm

Query	Precision (%)	Recall (%)	Accuracy (%)
Apple	94.03	95.18	94.61
Rose	94.21	95.46	94.84
Milk	93.81	95.24	94.53
Net	95.63	97.14	96.39
<b>Average</b>	<b>94.42</b>	<b>95.76</b>	<b>95.09</b>

To facilitate the comparison of the performance of the proposed Hybrid Semantic Algorithm, performances of the MFE\_CBIR, Hybrid Optimization Technique, and OntoRec were re-evaluated in the environment of the proposed system. The average performance of the chosen methodologies as well as the proposed Hybrid Semantic Algorithm is documented in Table 4. It is clearly inferable that the Hybrid Semantic Algorithm yields a better performance than all the systems used for comparison. The justification for a very high performance of the Hybrid Semantic Algorithm is that it uses APMI technique for semantic similarity computation and dynamically captures user intentions.

**Table 4** Comparison of performance of Hybrid Semantic Algorithm with other systems

Search technique	Average precision (%)	Average recall (%)	Accuracy (%)
MFE_CBIR [8]	83.4	80.6	82
Hybrid optimization technique [10]	88	88	88
OntoRec [11]	91.93	93.17	95.8
<b>Hybrid Semantic Algorithm</b>	<b>94.42</b>	<b>95.76</b>	<b>95.09</b>

## 6 Conclusions

Images are the most intrinsic part of the WWW pages in the most recent times [20]. Retrieving the most relevant image is a tedious task. A Hybridized Semantic Algorithm is proposed for Web image recommendation that incorporates Ontology modeling for homonyms and canonically synonymous ontologies. The proposed approach requires Ontology authoring and description for homonyms as well as synonymous ontologies. The semantic similarity is computed using the APMI strategy and also involves the construction of dynamic OntoPath based on the homonyms lookUp directory as well as Ontology classification through SVM. The proposed strategy also involves a user click feedback for various classes of images recommended. A strategic query expansion technique based on the users' choice in the OntoPath for a class of image as per users' intention is implemented. The proposed Hybrid Semantic Algorithm yields an average accuracy percentage of 95.09 which is much better than the existing Web page recommendation systems.

## References

1. Gordon, M., Pathak, P.: Finding information on the World Wide Web: the retrieval effectiveness of search engines. *Inf. Process. Manage.* **35**(2), 141–180 (1999)
2. Goodchild, M.F.: A spatial analytical perspective on geographical information systems. *Int. J. Geogr. Inf. Syst.* **1**(4), 327–334 (1987)
3. Ferrando, S.E., Doolittle, E.J., Bernal, A.J., Bernal L.J.: Probabilistic matching pursuit with gabor dictionaries. *Sig. Process.* **80**(10), 2099–2120 (2000)
4. Kanaegami, A., Koike, K., Taki, H., Ohgashi, H.: Text search system for locating on the basis of keyword matching and keyword relationship matching. US Patent 5,297,039 (1994)
5. Lang, K.: Newsweeder: learning to filter netnews. In: *Proceedings of the 12th International Conference on Machine Learning*, pp. 331–339 (1995)
6. Gong, Z., Cheang C.W.: Multi-term web Query Expansion using wordnet. In: *Database and Expert Systems Applications*, pp. 379–388. Springer, Berlin (2006)
7. Turney, P.: Mining the web for synonyms: PMI-IR versus LSA on TOEFL (2001)
8. Kousalya, S., Thananmani, A.S.: Image mining-similar image retrieval using multi-feature extraction and content based image retrieval technique. *Int. J. Adv. Res. Comput. Commun. Eng.* **2**(1), 4370–4372 (2013)
9. Dhonde, P., Raut, C.M.: Precise & proficient image mining using hierarchical K-means algorithm. *Int. J. Sci. Res. Publ.* **5**(1), 1–4 (2015)
10. Umaa Maheshvari, A., Thanushkodi, K.: Content based fast image retrieval using hybrid optimization techniques. In: *International Conference on Recent Advancements in Materials*. J. Chem. Pharm. Sci. 102–107 (2015)
11. Deepak, G., Andrade: OntoRec: a semantic approach for ontology driven web image search. In: *Proceedings of the International Conference on Big Data and Knowledge Discovery (ICBK)*, pp. 157–166 (2016)
12. Deng, Ke., Li, X., Lu, J., Zhou X.: Best keyword cover search. *IEEE Trans. Knowl. Data Eng.* **27**(1), 61–73 (2015)
13. Ma, Y., Wang, C., Jin, B.: A framework to normalize ontology representation for stable measurement. *J. Comput. Inform. Sci. Eng.* **15**(4) (2015)

14. Bedi, P., Thukral, A., Banati, H.: Focused crawling of tagged web resources using ontology. *Computers & Electrical Engineering*, vol. 39, no. 2, pp. 613–628. Elsevier (2013)
15. Sejal, D., Abhishek, D., Venugopal, K.R., Iyengar, S.S., Patnaik, L.M.: IR\_URFS\_VF: image recommendation with user relevance feedback session and visual features in vertical image search. *Int. J. Multimed. Infor. Retr.* **5**(4), 255–264 (2016)
16. Kalantidis, Y., Kennedy, L., Nguyen, H., Mellina, C., Shamma, D.A.: LOH and behold: web-scale visual search, recommendation and clustering using Locally Optimized Hashing. In: *European Conference on Computer Vision*, pp. 702–718. Springer International Publishing (2016)
17. Deepak, G., Priyadarshini, S.J.: Onto tagger: ontology focused image tagging system incorporating semantic deviation computing and strategic set expansion. *Int. J. Comput. Sci. Bus. Inform.* **16**(1) (2016)
18. Wang, M., Li, H., Tao, D., Ke, L., Xindong, W.: Multimodal graph-based re-ranking for web image search. *IEEE Trans. Image Process.* **21**(11), 4649–4661 (2012)
19. Chu, W.-T., Tsai, Y.-L.: A Hybrid Recommendation System Considering Visual Information for Predicting Favorite Restaurants. *World Wide Web*, pp. 1–19 (2017)
20. Shekhar, S., Singh, A., Agrawal, S.C.: An object centric image retrieval framework using multi-agent model for retrieving non-redundant web images. *Int. J. Image Min.* **1**(1), 4–22 (2015)

# Attribute Selection Based on Correlation Analysis



Jatin Bedi and Durga Toshniwal

**Abstract** Feature selection is one of the significant areas of research in the field of data mining, pattern recognition, and machine learning. One of the effective methods of feature selection is to determine the distinctive capability of the individual feature. More the distinctive capability the more interesting the feature is. But in addition to this, another important thing to be considered is the dependency between the different features. Highly dependent features lead to the inaccurate analysis or results. To solve this problem, we present an approach for feature selection based on the correlation analysis (ASCA) between the features. The algorithm works by iteratively removing the features that is highly dependent on each other. Firstly, we define the concept of multi-collinearity and its application to feature selection. Then, we present a new method for selection of attributes based on the correlation analysis. Finally, the proposed approach is tested on the benchmark datasets and the experimental results show that this approach works better than other existing feature selection algorithms both in terms of accuracy and computational overheads.

**Keywords** Feature selection · Correlation analysis · Multi-collinearity  
Attribute subset

## 1 Introduction

Data mining is the process of drawing useful patterns or information from the large database [1]. Feature selection is a data preprocessing technique that constitutes an important part of the knowledge discovery process [2]. Feature selection goals at

---

J. Bedi (✉) · D. Toshniwal  
Department of Computer Science & Engineering, Indian Institute of Technology,  
Roorkee 247667, India  
e-mail: jatinbedi278@gmail.com

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_5](https://doi.org/10.1007/978-981-10-7200-0_5)



**Fig. 1** Filter feature selection [4]



**Fig. 2** Wrapper feature selection [4]

finding the useful or relevant features from the data that encompass a number of irrelevant features in order to increase the accurateness of the process as well as to make the analysis process less time consuming. The purpose of the feature selection is to reduce the dimensionality of data while maintaining desired accuracy and also to solve the issues associated with the manual discovery process [3].

Feature selection models are ordered into three classes: filter, wrapper, and embedded. Hybrid model is a combination of the first two, i.e., filters and wrappers. Filter approaches are a forward selection approach that works in isolation from the learning algorithm [4] as contrary to the wrapper that works in conjunction with a learning algorithm [4] (Fig. 1).

The hybrid approach takes the benefits of both the approaches by combining them into one. The computational overhead associated with the wrapper is much more than of the filter due to the learning algorithm. Therefore, the present studies pay much more attention to the filter methods such as relief [5] and CFS [6] (Fig. 2).

The search strategy, generation of subsets, and the evaluation are the basic characteristics of the feature selection approaches [7]. Search strategy basically determines the way for searching the relevant features. It can be sequential, exhaustive, random, and heuristic search.

The subset generation step determines how the approach will proceed or the method for the selection of the features. Evaluation measure [7] is a set of functions used to assess the performance of the feature set selected.

This paper introduces an algorithm for feature selection. The algorithm is based on a method for solving the problem of multi-collinearity and works by iteratively removing the attribute based on the correlation analysis results.

The paper is organized into five sections including introduction. Section 2 provides a brief review of the work done in this field. The proposed approach for feature selection (ASCA) is described in Sect. 3. Section 4 discusses the experimental analysis of the proposed approach, and conclusion is stated in Sect. 5.

## 2 Literature Review

Feature selection aimed at finding out the subset of features from the entire set of features that represents the original features in an accurate manner. Feature selection has a number of applications in different fields including medical imaging [8], multimedia data [9], text classification [10], data mining [1], image processing. The problem of feature selection has been severe for a long time, particularly in the area of data mining. There exist a number of approaches for feature selection based on different methods used.

Chouchoulas and Shen [11] developed an algorithm for feature selection that works in a bottom-up manner and was based on the dependency metric. The algorithm starts with an empty set and progressively looks for an attribute to add to the set that cause highest increase in the metric.

Han et al. [12] introduced two algorithms for finding reducts based on the relative dependency measure that substitutes existing traditional dependency measure. One of the algorithms was based on the principle of backward elimination, while the entropy-based reduct was found using the second algorithm.

Shen and Chouchoulas [13] introduced the basic concepts related to using of rough set theory for attribute reduction or dimensionality reduction in complex systems and reviewed the various existing approaches for the feature selection. Deng presented the idea of the parallel reducts [14]. This new type of reduct is an extension of existing reduct, i.e., the Pawlak and dynamic reducts.

Liu and Setiono [15] developed a consistency subset evaluator algorithm based on the random searching. The algorithm works well by using random search to return the correct solutions. A correlation-based feature selection algorithm was introduced by Hall [2] that filters the search space to find the reducts based on the distinctive capabilities of the individual features.

Basak and Das [16] developed an approach for feature selection based on the correlation analysis and graph algorithm. The method works by comparing the correlation value with the average correlation and thus generating a graph on the basis of which the different features got selected.

Liu et al. [17] introduced the concept of the maximum nearest neighbor. The concept is based on how well a feature discriminates between the samples. The result of the application of this concept to benchmark dataset shows that it outperforms the various other existing feature selection approaches.

Ebrahimpour and Eftekhari [18] introduced a new method for feature selection called MRMR-MHS. It used the fuzzy set approach and is based on the maximum relevancy and minimum redundancy. The method works by selecting the attributes on the bases of ensemble ranking and inspired by correlation-based feature selection. Further, the experimental result over the benchmark dataset shows that this method works better than some of the previously existing approaches.

### 3 Proposed Method

#### 3.1 Problem of Multi-collinearity and Its Application to Feature Selection

Multi-collinearity [19] is basically a situation when two or more predictor variables are extremely related to each other; i.e., the value of one of the variables can be easily predicted by means of the value of the other variable with a desired level of accuracy. The correlation [19] between variables can take any value ranging from  $-1$  to  $1$ . If the correlation value is greater than  $0$ , then they are said to be positively correlated with each other; otherwise, if the value is less than zero, they are said to be negatively correlated with each other. There are various measures to detect the problem of multi-collinearity in the data for, e.g., tolerance factor, variable inflation factor.

The main problem associated with the multi-collinearity is that the analysis of such data becomes difficult to perform. There are numerous ways to solve the problem of multi-collinearity [19]. Correlation analysis is one of them to determine the set of the attributes that is highly correlated with each other. It works by dropping one of the two variables that is highly correlated with each other. The proposed approach makes use of this characteristic of the algorithm to generate an attribute set consisting of the set of attributes that are highly uncorrelated, i.e., the set of attributes lasting after the removal of linearity problem from the data.

The correlation value between the two attributes  $m$  and  $n$  is calculated by the formula [19]:

$$\text{Correl}(m, n) = \frac{\sum(m - \bar{m})(n - \bar{n})}{\sqrt{\sum(m - \bar{m})^2 \sum(n - \bar{n})^2}}$$

where  $\bar{m}$  and  $\bar{n}$  are the mean values.

#### 3.2 ASCA Algorithm

The algorithm works in a top-down manner. The method works by calculating the correlation [19] among the entire attribute in the dataset. The correlation value between them specifies how strongly the two attributes are related to each other. That further helps in the attribute reduction by exclusion of one of the attributes from the set that constitutes the highest correlation value.

**Algorithm:  $O(n^2)$** 

1. Initialize the entire dataset.
2. Compute the correlation value between set of attributes (remaining) in the dataset and store it in a table.
3. From the correlation table, identify the Attribute Set with the highest correlation value between them and denote it by  $\alpha$ .
  - (a) if  $\alpha > 0.4$ .
    - (i) Remove one of the two attributes (Attribute Set) from the dataset.
    - (ii) GOTO step 2.
  - (b) else
    - (i) Return the remaining attribute in the dataset as the reduced set.

**3.2.1 Example**

This section demonstrates the working of proposed approach with the help of an example. Consider the sample dataset given in Table 1a.

After doing correlation analysis over the attributes in the sample dataset, we get Table 1b as output. As per Table 1b, largest value of correlation is 0.83 (negatively correlated) that is greater than 0.4 and the significance value is also less than 0.05, so we randomly choose one out of the two attributes, i.e., either  $P1$  or  $P3$  and remove it from sample dataset. Table 2a shows the correlation result over the dataset obtained as output after the first iteration of ASCA, i.e., after removal of  $P1$ .

As per Table 2a, largest value of correlation is 0.71 (positively correlated) that is greater than 0.4 and the significance value is also less than 0.05, so we randomly choose one out of the two attributes, i.e., either  $P4$  or  $P5$  and remove it from sample dataset.

Table 2b shows the correlation result over the remaining dataset after the removal of  $P4$  attribute. According to correlation result in Table 2b, none of the attributes are correlated with each other; i.e., none of the value is greater than 0.4, so the final reduct set is:  $P2, P3, P5$ .

**4 Results and Discussion**

The ASCA algorithm and some well-known existing algorithm for feature selection such as CFSSubset [2] and PCA [21] were applied to the real-world Traffic Accident Casualty dataset [22].



**Table 1** a Sample dataset [20]. b Correlation results over sample dataset

a										b				
T/A	P1	P2	P3	P4	P5		P1	P2	P3	P4	P5			
F1	1	2	0	1	1	P1	1	-0.059	<b>-0.830**</b>	-0.459	-0.459			
F2	1	2	0	1	1		Correlation	0.872	<b>0.003</b>	0.182	0.182			
F3	2	0	0	1	0	P2	-0.059	1	-0.154	-0.128	0.311			
F4	0	0	1	2	1		Correlation	0.872	0.670	0.724	0.382			
F5	2	1	0	2	1	P3	<b>-0.830**</b>	-0.154	1	0.603	0.302			
F6	0	0	1	2	2		Correlation	0.670		0.065	0.397			
F7	2	0	0	1	0	P4	-0.459	-0.128	0.603	1	0.714*			
F8	0	1	2	2	1		Correlation	0.724	0.065		0.020			
F9	2	1	0	2	2	P5	-0.459	0.311	0.302	0.714*	1			
F10	2	0	0	1	0		Correlation	0.182	0.397	0.020				
							Sig. (2-tailed)	0.382						

**Table 2** a Correlation results after removal of P1 (after first iteration). b Correlation results after removal of P4 (after second iteration)

		a					b				
		P2	P3	P4	P5		P2	P3	P4	P5	
P2	Correlation	1	-0.154	-0.128	0.311						
	Sig. (2-tailed)	-	0.670	0.724	0.382	P2	Correlation	1	-0.154	0.311	
P3	Correlation	-0.154	1	0.603	0.302		Sig. (2-tailed)	-	0.670	0.382	
	Sig. (2-tailed)	0.670	-	0.065	0.397	P3	Correlation	-0.154	1	0.302	
P4	Correlation	-0.128	0.603	1	<b>0.714*</b>		Sig. (2-tailed)	0.670	-	<b>0.714*</b>	
	Sig. (2-tailed)	0.724	0.065	-	<b>0.020</b>	P5	Correlation	0.311	0.302	1	
P5	Correlation	0.311	0.302	<b>0.714*</b>	1		Sig. (2-tailed)	0.382	0.397	0.397	
	Sig. (2-tailed)	0.382	0.397	<b>0.020</b>	-						

## 4.1 Dataset Description

No. of attribute	Attributes type	No. of instances
16	Numeric	1,86,190

### Attribute:

1. Accident-Index
2. Vehicle-Reference
3. Casualty-Reference
4. Casualty-Class
5. Sex-of-Casualty
6. Age-of-Casualty
7. Age-Band-of-Casualty
8. Casualty-Severity
9. Pedestrian-Location
10. Pedestrian-Movement
11. Car-Passenger
12. Bus-or-Coach-Passenger
13. Pedestrian-Road-Maintenance-Worker
14. Casualty-Type
15. Casualty-Home-Area-Type
16. Casualty-IMD-Decile.

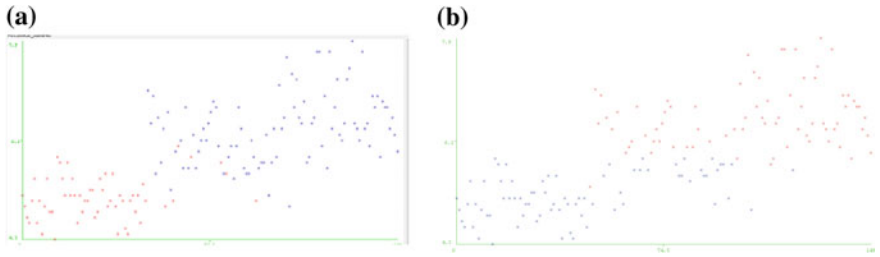
The reduced dataset obtained as result is then clustered using the simple k-means algorithm [1], and Within Cluster Sum of Squared Errors have been compared; it is found that the proposed method produces better results than that of CFSubset [2] approach and PCA [21] approach works better than that of the proposed approach. But the computational overhead associated with the PCA [21] algorithm is much

**Table 3** Comparison results

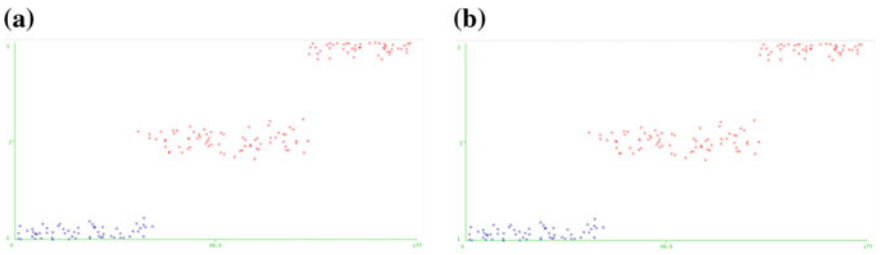
Dataset	Feature selection technique	k-means clustering (Within Cluster Sum of Squared Errors)
IRIS	ASCA	3.81
	CFSubset	4.13
	PCA	3.0
	Rf-Nw	4.44
WINE	ASCA	19.27
	CFSubset	19.275
	PCA	12.91
	Rf-Nw	22.07
ACCIDENT CAUSALITY	ASCA	14768.36
	CFSubset	33586.63

higher than that of the proposed approach. So in terms of computational overhead associated the proposed approach is better than PCA [21].

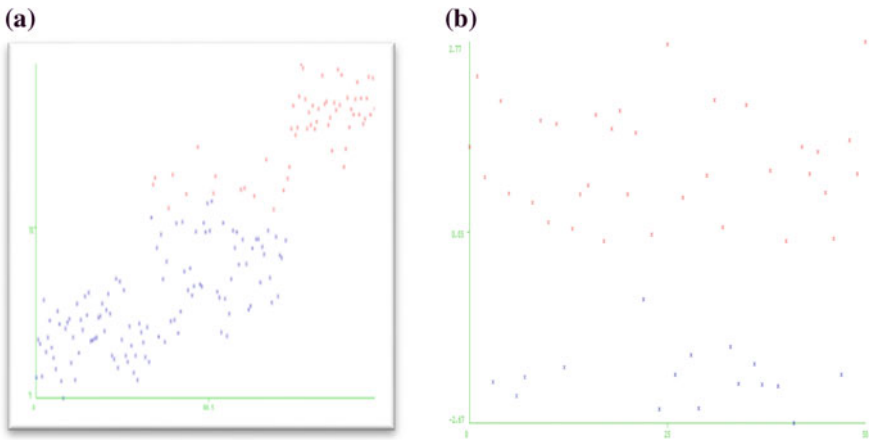
In addition to this, the ASCA algorithm is also applied over the wine dataset, Iris dataset collected from the UCI repository [23] and the reduced set obtained as an



**Fig. 3** a Clustering result using CFSubset (IRIS DATASET). b Clustering result using ASCA



**Fig. 4** a Clustering result using CFSubset (WINE DATASET). b Clustering result using ASCA



**Fig. 5** a Clustering result using PCA (WINE DATASET). b Clustering result using PCA (IRIS DATASET)

output is clustered by k-Means Clustering algorithm. Within Cluster Sum of Squared Errors are compared, and results are listed in Table 3, which demonstrate that ASCA produces superior result than CFSubset. Figures 3, 4, and 5 show the clustering result (over reduced set) generated using ASCA, CFSubset [2], and PCA [21] algorithms.

## 5 Conclusion

The datasets consist of a large number of attributes, and some of the attribute values may be highly correlated with each other. So, it becomes difficult to get the accurate result based on the analysis of the data. Feature selection can serve as a method to reduce the dimensionality by considering the different alternative attribute sets depending on the method used to make the analysis more precise. This paper presents an approach for attribute selection based on the correlation analysis to return the attribute set that consists of highly uncorrelated attributes. Further, the experimental result over the various datasets shows that this approach works better than the other approaches. The computational overhead associated with the approach is much less than other existing algorithms. But the main problem with the approach is that it can be applied only to the numeric dataset.

## References

1. Pei, H.K.: *Data Mining: Concepts and Techniques*, 3rd edn. Elsevier (2011)
2. Hall, M.: *Correlation-Based Feature Subset Selection for Machine Learning*. University of Waikato, Hamilton, New Zealand (1998)
3. Fayyad, U., Piatetsky, G., Smyth, P.: From data mining to knowledge discovery in databases. *AI Magazine* (1996)
4. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Comput. Electr. Eng.* **40**, 14–28 (2014)
5. Kononenko, I.: Estimation attributes: analysis and extensions of RELIEF. In: *European Conference on Machine Learning*, New Brunswick (1994)
6. Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning. In: *17th International Conference on Machine Learning* (2000)
7. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. *Artif. Intell.* **97**(1), 245–271 (1997)
8. Cheng, T.-H., Wei, C.P., Tseng, V.: Feature selection for medical data mining. In: *IEEE International Symposium on Computer-Based Medical Systems*, pp. 165–170 (2006)
9. Rahman, M.N.A., Lazim, Y.M., Mohamed, F.: Applying rough set theory in multimedia data classification. *Int. J. New Comput. Archit Appl* 683–693 (2011)
10. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: *Proceedings of the 14th International Conference on Machine Learning* (1997)
11. Chouchoulas, A., Shen, Q.: Rough set-aided keyword reduction for text categorisation. *Appl. Artif. Intell.* **15**(9), 843–873 (2001)

12. Han, J., Hu, X., Lin, T.Y.: Feature subset selection based on relative dependency between attributes. In: 4th International Conference Rough Sets and Current Trends in Computing, Uppsala, Sweden (2004)
13. Shen, Q., Chouchoulas, A.: A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems. *Eng. Appl. Artif. Intell.* 263–278 (2002)
14. Deng, D.: Parallel reduct and its properties. In: *Granular Computing*, pp. 121–125 (2009)
15. Liu, H., Setiono, R.: A probabilistic approach to feature selection—a filter solution. In: 13th International Conference on Machine Learning (1996)
16. Basak, A., Das, A.K.: A graph based feature selection algorithm utilizing attribute intercorrelation. In: *IEEE* (2016)
17. Liu, J., et al.: Feature selection based on quality of information. *Neurocomputing* **225**, 11–22 (2017)
18. Ebrahimipour, M.K., Eftekhari, M.: Ensemble of feature selection methods: a hesitant fuzzy sets approach. *Appl. Soft Comput.* 300–312 (2017)
19. Gujarati, D.: *Basic Econometrics*, 4th edn. McGraw-HILL, USA (1995)
20. Wikipedia. [http://en.m.wikipedia.org/wiki/Rough\\_set](http://en.m.wikipedia.org/wiki/Rough_set). Accessed July 2016
21. Mozer, M.C., Jordan, M.I., Petsche, T.: A principled alternative to the self-organising map. In: *Advances in Neural Information Processing Systems*, Cambridge (1997)
22. DATA.GOV.UK. <https://data.gov.uk/dataset/road-accidents-safety-data>. Accessed 01 Nov 2016
23. Murphy, P., Aha, W.: UCI repository of machine learning databases (1996). <http://www.ics.uci.edu/mllearn/MLRepository.html>

# Taxi Travel Time Prediction Using Ensemble-Based Random Forest and Gradient Boosting Model



**Bharat Gupta, Shivam Awasthi, Rudraksha Gupta, Likhama Ram, Pramod Kumar, Bakshi Rohit Prasad and Sonali Agarwal**

**Abstract** Proposed work uses big data analysis and machine learning approach to accurately predict the taxi travel time for a trip based on its partial trajectory. To achieve the target, ensemble learning approach is used appropriately. Large dataset used in this work consists of 1.7 million trips by 442 taxis in Porto over a year. Significant features are extracted from the dataset, and Random Forest as well as Gradient Boosting is trained on those features and their performance is evaluated. We compared the results and checked the efficiency of both in this regard. Moreover, data inferences are done for trip time distribution, taxi demand distribution, most traversed area, and trip length distribution. Based on statistics, errors, graphs, and results, it is observed that both the methods predict time efficiently, but Gradient Boosting is slightly better than Random Forest.

**Keywords** Taxi travel time · Ensemble · Random Forest · Gradient Boosting

---

B. Gupta (✉) · S. Awasthi · R. Gupta · L. Ram · P. Kumar · B. Rohit Prasad · S. Agarwal  
Indian Institute of Information Technology, Allahabad, India  
e-mail: iit2014156@iiita.ac.in

S. Awasthi  
e-mail: iit2014155@iiita.ac.in

R. Gupta  
e-mail: iit2014158@iiita.ac.in

L. Ram  
e-mail: iit2014163@iiita.ac.in

P. Kumar  
e-mail: iit2014082@iiita.ac.in

B. Rohit Prasad  
e-mail: rohit.cs12@gmail.com

S. Agarwal  
e-mail: sonali@iiita.ac.in

## 1 Introduction

The proposed work involves travel time prediction for taxi cabs in a city based on historic data. It utilizes an ensemble learning approach to predict travel time. It consists of 3 main steps—pre-processing, supervised learning, and prediction. Random Forest and Gradient-Boosted Trees are used. Their results and performance are evaluated. Initially, a partial trajectory with its initial and final location is provided which the taxi has traversed, and then, the time it will take to reach the final destination from the current location of the taxi is predicted using ensemble approach. Currently, the electronic dispatch system gathers the geo-locations of different taxis at different point in time to provide the information of availability of taxis at the booking location. We can use information about the time stamp and the geo-locations as a valuable source of information for prediction of stipulated traveling times, the demand distribution, and traffic pattern at different locations which is the objective of the proposed work.

Paper is organized into five sections. Section 1 introduces taxi prediction based on time stamps and geo-locations measured by current electronic dispatch system. Section 2 provides an extensive survey. In Sect. 3, authors have explained the detailed methodology and techniques adopted in the proposed approach. Experimental results and its comprehensive analysis are given in Sect. 4. Finally, Sect. 5 concludes the findings of the authors work and provides future scope directions.

## 2 Related Work

The research article [1] specifies the applicability of ensemble learning and trip matching for estimating the destination and trip time for taxis in real time using Haversine distance calculated by Kernel Regression which are used as features to estimate the final destination, combined with average speed, average acceleration, and shape complexity for trip time prediction. Another work [2] partitions input into different clusters using K-Means and membership degree to the cluster centers which are measured by “Gaussian fuzzy membership function.” Authors in their work [3] tried to make passengers pick up profitable by assigning a score to each road segment that determines whether it should be picked in the cruising route or not. This score depends on several factors; probability of finding a passenger on a road segment, length of the occupied paths originating from the road segment, etc. Research work [4] took a large number of trips, and a probabilistic model is devised to detect parking places for taxis. Also, the intelligent guesses made by passenger recommender have been enhanced by estimating the waiting time on a road segment. Research work [5] established a method to identify taxi’s cruising route as well as travel time calculation



prepares a “Link Cost Table (LCT)” using input dataset which identifies the cruising routes. The travel time calculation method is developed using LCT and Dijkstra algorithm. The aim of work in [6] is to predict travel time from GPS measurements of probe vehicles. A probabilistic model based on expectation-maximization algorithm is used. Work in [7] uses automatic vehicle location data which stream through three components—tracker, filter, and predictor—which transform the data and give the predicted result of arrival/departure. Authors of research article [8] showed that traffic flow data and its quick analysis are very essential for which they applied Microsoft Time Series (MTS) algorithm. Approach discussed in work [9] does travel time prediction by combining multivariate regression, principal component analysis, k-nearest neighbors (KNN), cross-validation, and exponentially weighted moving average (EWMA) methods. Better results were found by “KNN” method. As discussed in article [10], AVL is used to track and monitor a computer transport in real life. This work presents global positioning system (GPS)-based AVL system for bus transit. In study presented in [11], authors focus on huge-gauge-geo-position data with fractional info and build a proper path to calculate the possibility of a route being taken by the taxicab driver. The research work [12] exploits this fact to the fullest for route prediction by matching the initial journey of the driver’s trip with one of the sets of previously observed trips from the database. The research in [13] introduces a “temporal probability grid network” where each grid has two properties: probability and capacity. Prime focus in [14] is building a predictive framework to estimate the endpoint, and the total cruising time of taxis based on their initial partial trips is the idea behind the competition. A two-stage modeling approach is used in [15] to predict movements of vacant taxis in search of customers. The first one predicts the preference of vacant taxi drivers, and second one finds the distance and circulation time of vacant taxi drivers by their customer preference decisions predicted by first stage sub-model in their given zone.

### 3 Proposed Methodology

#### 3.1 Dataset Description

Presented work uses a dataset of 442 cabs and their trips in the city of Porto for a period of 1 year. The dataset contains 1.7 million entries. Trip’s trajectory is sampled at time interval of 15 s. The description of the dataset fields is given in Table 1.

After doing the literature review of significant research work, we have prepared a brief overview of different processing steps that we will be following according to our plan of work.

**Table 1** Dataset description

S.No.	FAL range	Description/Values
1	TRIP_ID	It uniquely identifies each trip
2	CALL TYPE	online/on road
3	ORIGIN CALL	The caller's identity
4	ORIGIN STAND	Taxi stand identifier
5	TAXI ID	ID of each taxi
6	TIMESTAMP	Starting time of a trip
7	DAYTYPE	Type of the day (Holiday, Day before Holiday, Weekday)
8	MISSING DATA	FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing
9	POLYLINE	The array of trip's locations (GPS locations)

### 3.2 Pre-processing—Feature Selection

Initially, we have a partial trajectory with its initial and final location which the taxi has traversed. We are extracting some key features from the dataset to train the Random Forest and Gradient Boosting which are source latitude and longitude (from POLYLINE), destination latitude/longitude (from POLYLINE), average speed (from POLYLINE), week day (from TIMESTAMP), hour (from TIMESTAMP), and day type (from DAY TYPE). To calculate the average speed, we need the distance between each point in the trip. To calculate the distance between two points, we have used ‘‘Haversine Formula’’ using their latitude and longitude.

$$\begin{aligned} \text{hav}\left(\frac{d}{r}\right) &= \text{hav}(\varnothing_2 - \varnothing_1) + \cos \varnothing_2 \cos \varnothing_1 (\lambda_2 - \lambda_1) \\ \text{hav}(\theta) &= \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos \theta}{2} \end{aligned} \quad (1)$$

where  $d$ : the distance between the two points (along a great circle of the sphere),  $r$ : the radius of the sphere,  $\varnothing_1$  and  $\varnothing_2$ : latitude of point 1 and latitude of point 2,  $\lambda_1$ ,  $\lambda_2$ : longitude of point 1 and longitude of point 2.

### 3.3 Supervised Learning

The goal of supervised learning is to approximate the mapping function so well that when a new input data  $x$  is given, the output variable  $Y$  can be predicted for that data. Since we have to predict ‘‘time’’ which is a continuous-valued attribute, we will be using Regression Technique; Random Forest Regression and Gradient

Boosting are used to predict the travel time of a taxi based on the partial trajectory. These techniques are briefed in following subsections.

### 3.4 Decision Tree

A decision tree is made by splitting on each feature for a random number of samples. The split is made on that feature whose standard deviation reduction is maximum.

(1) *Random Forest Regression*: Multiple forests are grown rather a single tree. One sample from input is taken at random with replacement. Tables are drawn for each independent variable resulting in the dependent variable. Calculations are then made for specific cases where probabilities are taken for each attribute and the average is then considered to predict the final result. The final output is the maximum voted result or average of each tree. In Python, command for Random Forest function is as below:

```
clf = ensemble.RandomForestRegressor()
```

Important parameters to this function are `n_estimators`—# of trees considered in the forest, `random_states`—the random seed for the random number generator, and `n_jobs`—# of cores to be used for training the dataset.

### 3.5 Gradient Tree Boosting

Boosting refers to family of algorithms that convert weak learners to strong learners. Several weaker rules are clubbed together to form stronger rules that can make generalizations. Conversion is done by average or weighted mean, in an iterative fashion. An initial model is predicted using a loss function. Each time a decision tree is generated and model is updated based on previous model and loss function resulting in a final model. In Python, command for Gradient Boosting function is given below:

```
clf = ensemble.GradientBoostingRegressor()
```

Important parameters to this function are `n_estimators`—# of trees considered in the forest, `random_states`—the random seed for the random number generator `max_depth`—the maximum depth up to which the tree is generated.

## 4 Results and Discussion

Experiments are performed to assess the impact of different significant control parameters over the performance specified in subsequent subsections.

### 4.1 Variation in Parameters

- (1) *Graph for 1000 entries:* Each line in the graph shown in Fig. 1 is plotted for variations in standard deviation (SD) according to increase in number of estimators for a particular constant random state. Similarly, each line in the graph shown in Fig. 2 is plotted for variations in standard deviation according to increase in number of random states for a particular value of number of estimators.

Graph in Figs. 1 and 2 tells us that optimum value of parameters Random States and Estimators should be 19 and 300, respectively, for Random Forest Regression to have a minimum value of standard deviation. Similarly, graphs in Figs. 3 and 4 tell us that optimum value of parameters Random States and Max Depth should be 19 and 11, respectively, for Gradient Boosting to have a minimum value of standard deviation.

- (2) *Graphs for 3000 entries:* Graphs plotted in Figs. 5 and 6 specify that the optimum value of the parameters Random States and Estimators should be 21

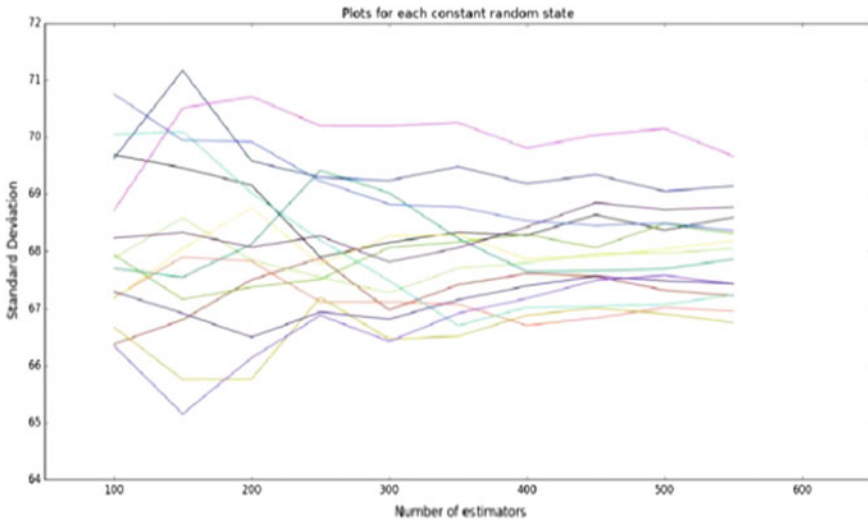


Fig. 1 SD plot for estimators and constant random states

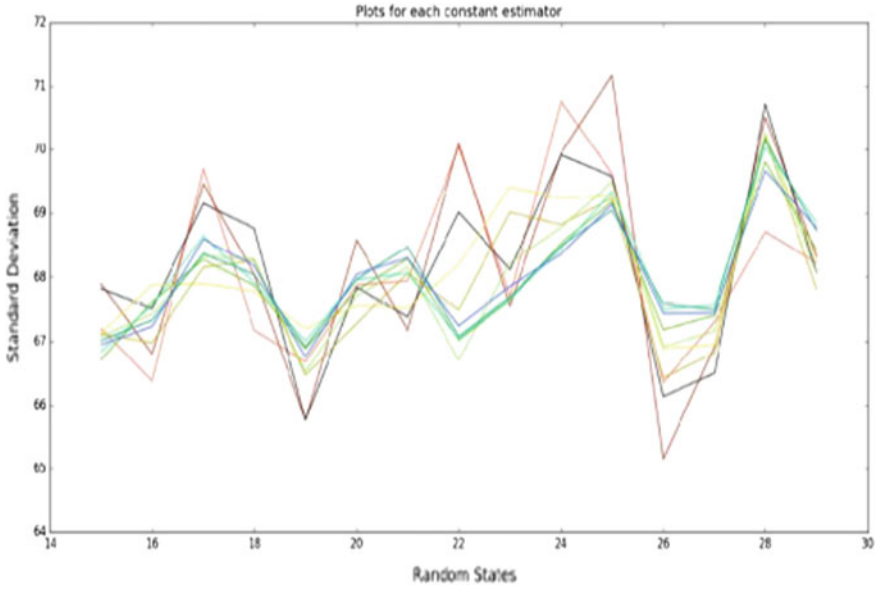


Fig. 2 SD plot for random states and constant estimators

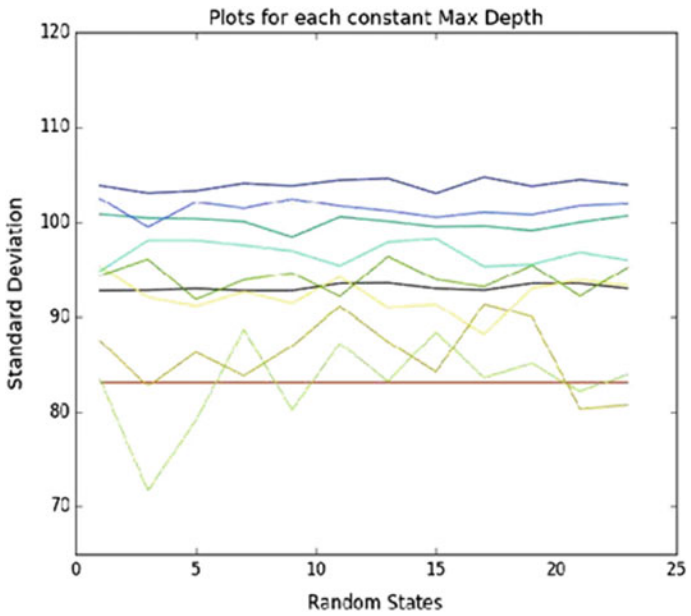


Fig. 3 SD plot for random states and constant depths

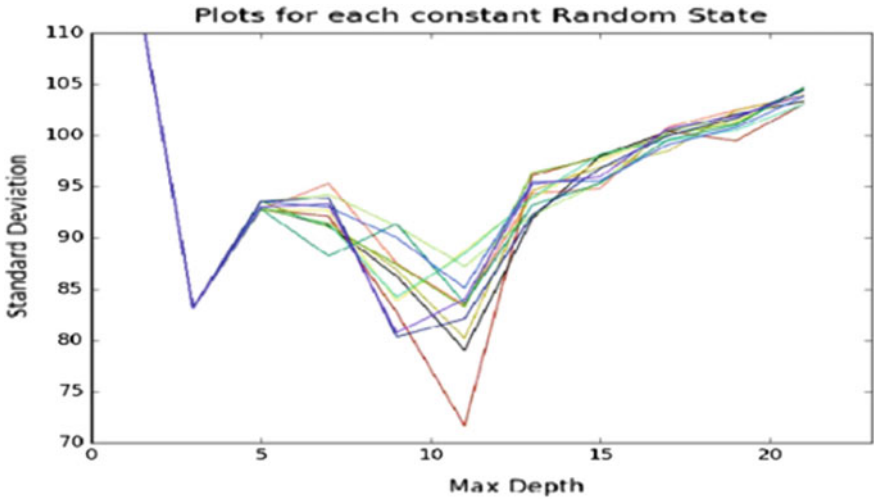


Fig. 4 SD plot for max depth and constant random states

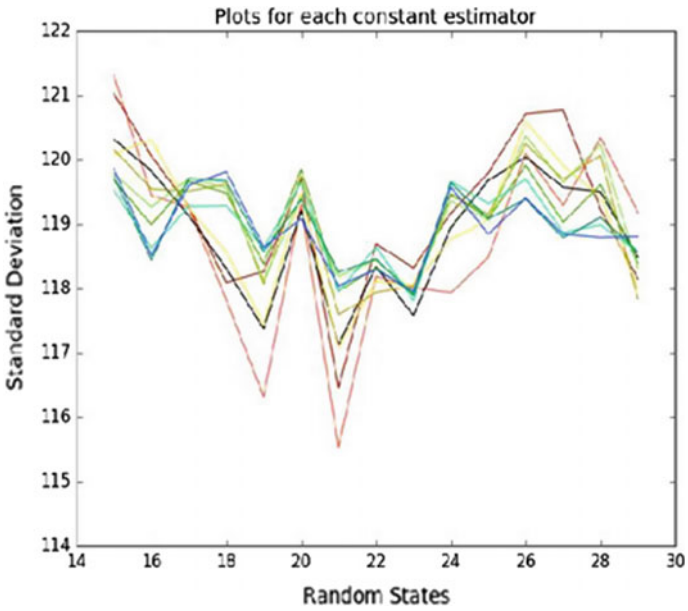


Fig. 5 SD plot for random states and constant estimators with 3000 instances

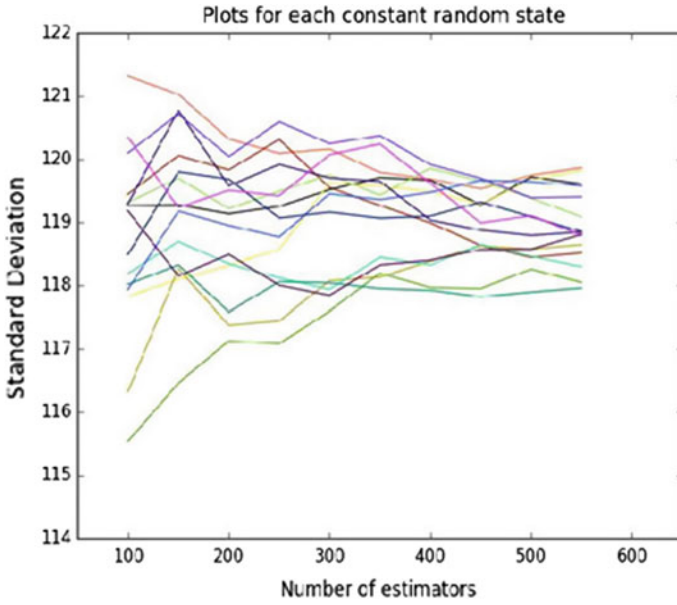


Fig. 6 SD plot for estimators and constant random states with 3000 instances

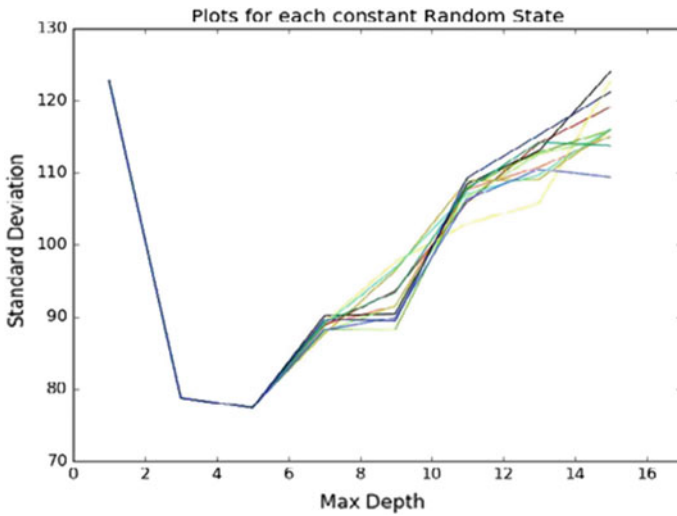
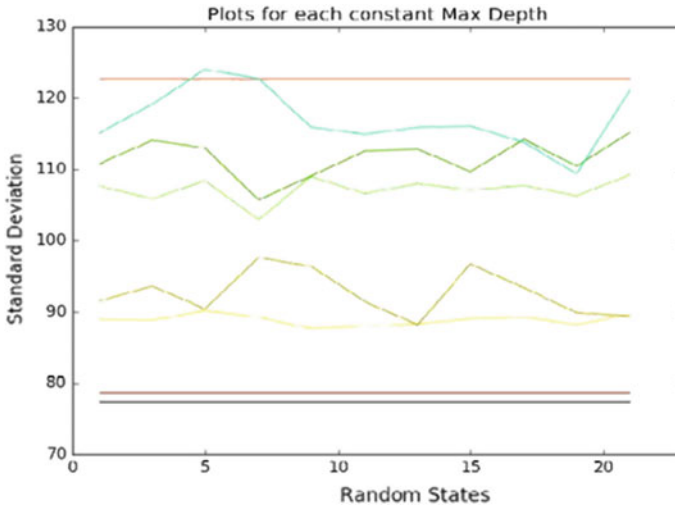


Fig. 7 SD plot for max depths and constant random states with 3000 instances



**Fig. 8** SD plot for random states and constant max depths with 3000 instances

and 300, respectively, for Random Forest Regression to have a minimum value of standard deviation. Graphs shown in Figs. 7 and 8 establish the fact that the optimum value of the parameters Max Depth and Random States should be 5 and 19, respectively, for Gradient Boosting to have a minimum value of standard deviation.

- (3) *Graphs for 5000 entries*: Graph plotted in Figs. 9 and 10 depicts that optimum value of parameters Random States and Estimators should be 20 and 400, respectively, for Random Forest Regression to have a minimum value of standard deviation.
- (4) *Graphs for 10000 entries*: Figures 11 and 12 specify that optimum value of the parameters Random States and Estimators should be 21 and 350, respectively, for Random Forest Regression to have a minimum value of standard deviation.

## 4.2 Data Inference

In addition to the performance evaluation and optimization, this work aims to make the certain significant statistical inferences from the traffic dataset. Figure 13 shows the number of trips per one hour interval of a day which is helpful in assessing the traffic load in a certain interval. On the other hand, trip length and trip time distribution over 1440 minutes duration in a day is shown in Figs. 14 and 15, respectively.



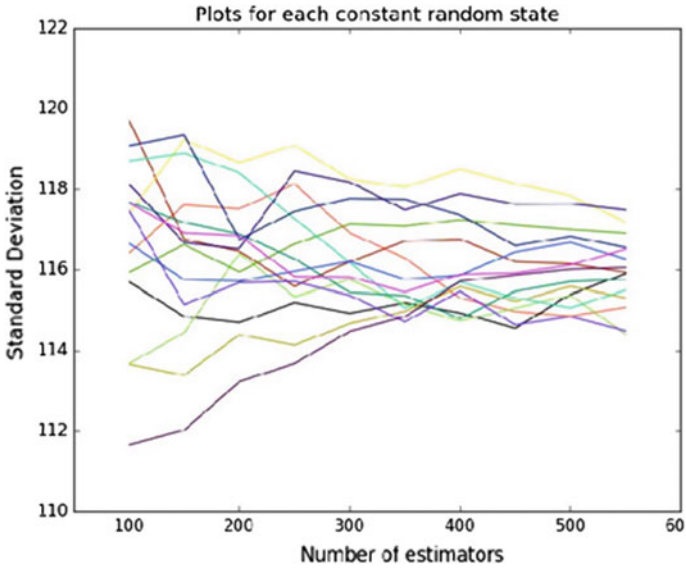


Fig. 9 SD plot for estimators and constant random states with 5000 instances

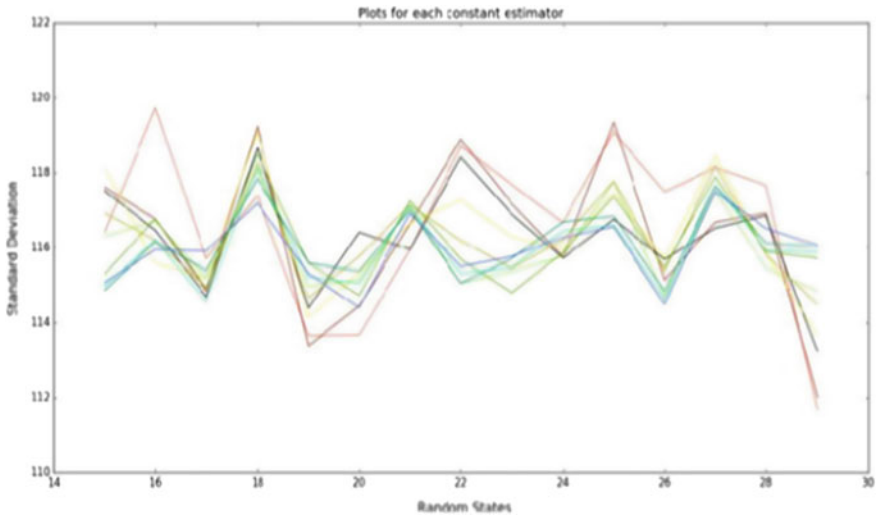


Fig. 10 SD plot for random states and constant estimators with 5000 instances

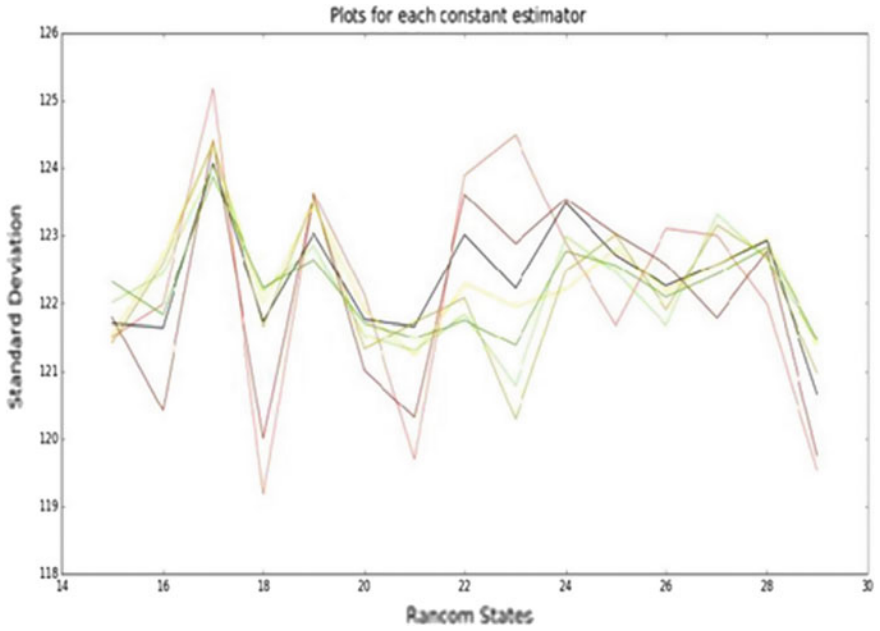


Fig. 11 SD plot for random states and constant estimators with 10000 instances

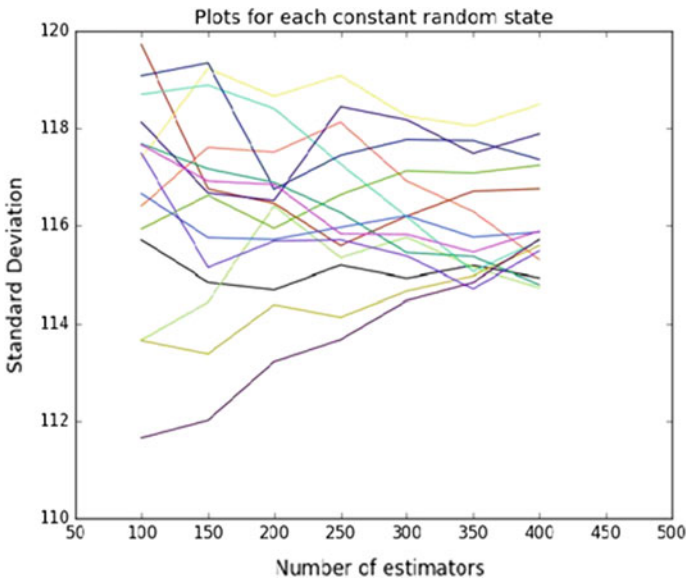


Fig. 12 SD plot for estimators and constant random states with 10000 instances

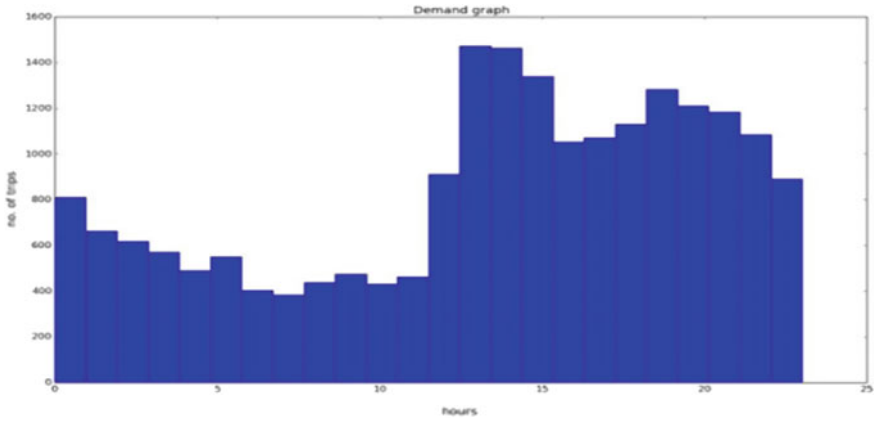


Fig. 13 Demand distribution of taxi along daytime

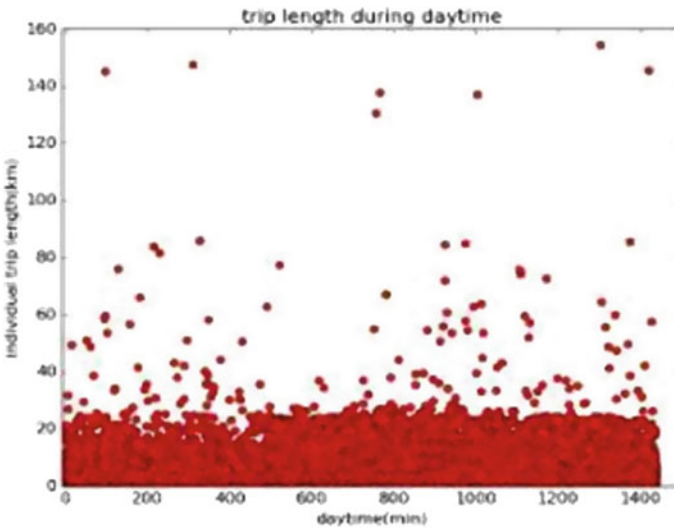


Fig. 14 Trip length distribution in a period of 1440 min in 1 day

### 4.3 Performance Evaluation

In current work, performance of different models is measured using Root Mean Square Logarithmic Error (RMSLE), Root Mean Squared Error (RMSE), Mean Absolute error (MAE), and Mean Relative Error (MRE).

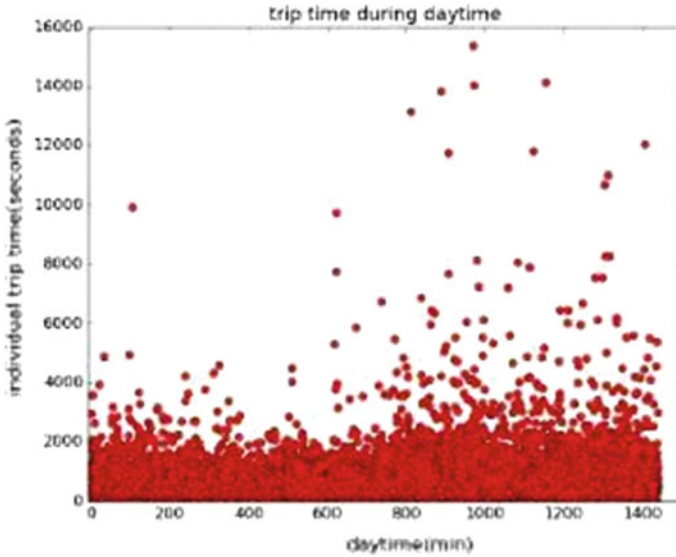


Fig. 15 Trip time distribution in a period of 1440 min (1 day)

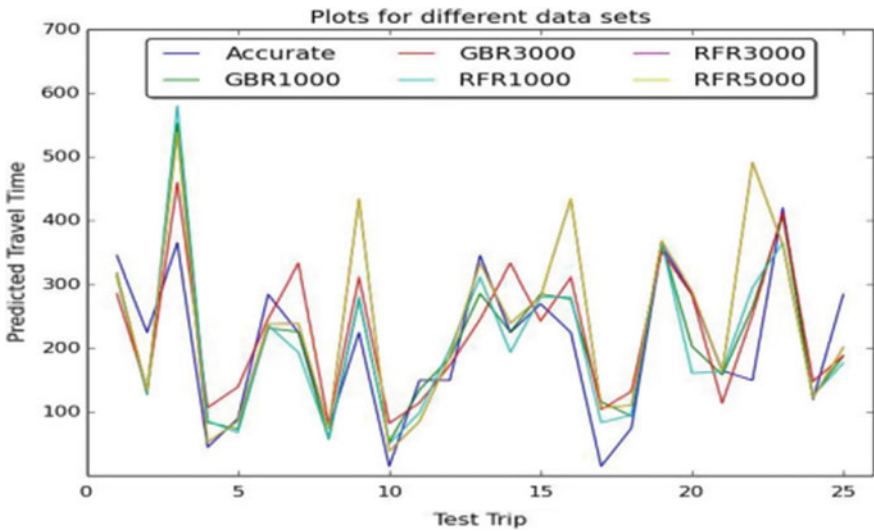


Fig. 16 Prediction time plot

As per the results shown in Fig. 16 for prediction time and Table 2 for performance measurements, it is evident that Gradient Boosting Regression predicts better results than Random Forest Regression. Also, it was noted that with increase

**Table 2** Performance comparison

Methods/Errors	RMSE	RMSLE	MAE	MRE	SD
Random Forest Regression on 1000	92.26	0.54	59.79	0.29	91.89
Gradient Boosting Regression on 1000	83.60	0.56	51.79	0.24	82.62
Random Forest Regression on 3000	114.87	0.58	67.62	0.29	108.88
Gradient Boosting Regression on 3000	81.02	0.62	64.87	0.29	77.42
Random Forest Regression on 5000	46.66	0.43	35.15	0.17	46.62
Random Forest Regression on 10000	46.51	0.38	28.84	0.28	52.30

**Table 3** Experimental setting

	Setting 1	Setting 2	Setting 3
Training set	Without outliers	With outliers	With outliers
Testing set	Without outliers	Without outliers	With outliers

**Table 4** Correlation and RMSE statistics

	Correlation	RMS Error (s)
Method 1	0.863	193.570
Method 2	0.826	216.110

in data size, results become more favorable. Moreover, we compared the results with that of quoted in other research works [5, 9] in similar problem domain and experimental settings. It is found that presented approach in current work gives better results as evident from Tables 2, 3, and 4.

## 5 Conclusion

It is found that Gradient Boosting Regression predicts better results than Random Forest Regression. However, Random Forest Regression works faster than Gradient Boosting Regression as it uses parallel processing for making trees. Moreover, with increase in training data size, prediction improves further. Also, with increase in training data size, there is little variation in `n_estimators` and `random_states` parameters of Random Forest Regression for result improvement. In case of Gradient Boosting Regression with increase in training data size, there is no variation in `random_states` and `max_depth`, and `n_estimators` decrease. We intend to run our algorithms on the complete dataset to further analyze the outcomes. We also wish to implement it in the form of an android or Web application for the convenience of the passengers booking cabs. Cab operators can use this system to improve the efficiency of the electronic dispatch system of the taxis.

## References

1. Lam, H.T., Diaz-Aviles, E., Pascale, A., Gkoufas, Y., Chen, B.: Taxi destination and trip time prediction from partial trajectories. *IBM Research—Ireland* (2015)
2. Tang, J., Zou, Y., Ash, J., Zhang, S., Liu, F., Wang, Y.: Travel time estimation using freeway point detector data based on evolving fuzzy neural inference (2016)
3. Dong, H., Zhang, X., Dong, Y., Chen, C., Rao, F.: Recommend a profitable cruising route for taxi drivers (2014)
4. Yuan, N.J., Zheng, Y., Zhang, L., Xie, X.: T-Finder: a recommender system for finding passengers and vacant taxis (2013)
5. Miwa, T., Sakai, T., Morikawa, T.: Route identification and travel time prediction using probe-car data (2004)
6. Hunter, T., Herring, R., Abbeel, P., Bayen, A.: Path and travel time inference from GPS probe vehicle data (2009)
7. Cathey, F.W., Dailey, D.J.: A prescription for transit arrival/departure prediction using automatic vehicle location data. *Trans Res Part C: Emerg Technol* **11**(3–4) (2013)
8. Luhang, X.: The research of data mining in traffic flow data. Shandong University (2015)
9. Cavar, I., Kavran, Z., Bosnjak, R.: Estimation of travel times on signalized arterials (2013)
10. Oluwatobi, A.N.: A GPS based automatic vehicle location (AVL) system for bus transit (2014)
11. Zhan, X., Hasan, S., Ukkusuri, S.V., Kamga, C.: Urban link travel time estimation using large-scale taxi data with partial information (2013)
12. Froehlich, J., Krumm, J.: Route prediction from trip observations (2008)
13. Yang, W.: Recommending Profitable Taxi Travel Routes based on Big Taxi Trajectory Data (2015)
14. Hoch, T.: An ensemble learning approach for the Kaggle taxi travel time prediction challenge (2015)
15. Wong, R.C.P., Szeto, W.Y., Wong, S.C.: A two-stage approach to modelling vacant taxi movements (2015)

# Virtual Machine Migration—A Perspective Study



Christina Terese Joseph, John Paul Martin, K. Chandrasekaran  
and A. Kandasamy

**Abstract** The technology of Cloud computing has been ruling the IT world for the past few decades. One of the most notable tools that helped in prolonging the reign of Cloud computing is virtualization. While virtualization continues to be a boon for the Cloud technology, it is not short of its own pitfalls. One such pitfall results from the migration of virtual machines. Though migration incurs an overhead on the system, an efficient system cannot neglect migrating the virtual machines. This work attempts to carry out a perspective study on virtual machine migration. The various migration techniques proposed in the literature have been classified based on the aspects of migration that they consider. A survey of the various metrics that characterize the performance of a migration technique is also done.

**Keywords** Cloud computing · Virtual machines (VM) · Migration

## 1 Introduction

Cloud is no longer a “new” technology. Having its roots in networks, which exists since the 1960s, the term Cloud has been around for quite sometime now. To the question of why the term “Cloud,” the answer is simple: One of the features of Cloud in nature is that it can change shape easily. In a similar manner, the Cloud also provides “elasticity” where it can change its size according to the requirements. Another reason which contributed to the term coining is that networks were usually

---

C. T. Joseph (✉) · K. Chandrasekaran  
Department of Computer Science and Engineering, National Institute of Technology,  
Surathkal 575025, Karnataka, India  
e-mail: xtina1232@gmail.com

J. P. Martin · A. Kandasamy  
Department of Mathematical and Computational Sciences, National Institute  
of Technology, Surathkal 575025, Karnataka, India

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_7](https://doi.org/10.1007/978-981-10-7200-0_7)

represented as clouds. Cloud computing, with its attractive features, has grabbed the interest of many users. While some users intentionally make use of Cloud, other users get the advantages of Cloud while they remain unaware of it, for instance, when they make use of the storage provided by e-mail providers such as the Google Drive provided by Gmail, OneDrive provided by Outlook. Social networking sites such as Facebook, Twitter are also based on Cloud.

One of the key concepts of the Cloud system is “**Virtualization.**” Virtualization can be explained in layman terms as something similar to an apartment. In an apartment, a single building is divided or partitioned into different components which serve as homes to different individuals or families. Each family feels that the home is their own. Similarly in Cloud, virtualization is creating virtual entities. In cloud, the same physical machine may run one or more virtual machines (VMs). The virtual machines are monitored and managed by a layer on top of the hardware called “hypervisors” also called virtual machine monitors (VMMs).

In this study, we attempt to look at the existing works on virtual migration, from a new angle. Section 2 describes some of the metrics used in the literature to assess the different policies for VM migration. Section 3 provides a taxonomy and discusses few works in each category. Section 4 includes some open research problems. Finally, the paper is concluded in Sect. 5.

## 2 Virtual Machine Migration

The word migration refers to the movement of an object from one place to another. In virtual machine migration, it is the virtual machine that is in motion. VM migration is the migration of virtual machines across physical machines through the network. It involves transfer of the process state, memory contents, and other elements related to the virtual machine from one physical machine to a different machine.

### 2.1 Metrics Used for VM Migration

VM migration policies are defined to determine which machine is to be migrated, whether migration should be carried out at an instant of time, and also the destination of the migrated VM. A summary of parameters obtained from the study of related works is given in Table 1. Consolidating the information in the table into a pie-chart in Fig. 1, it can be seen that total migration time and downtime are the most common metrics.



**Table 1** VM migration metrics used by researchers

Sl. no.	Author/s	Metric used	Metric description (if any)
1	Huang et al. [1]	Total migration time	Time taken to migrate
		Migration downtime	Time during which the service of the VM is not available
		Network contentions	Contention on network bandwidth
2	Huang et al. [2]	Downtime	The time during which the migrating VMs execution is stopped
		Total migration time	The period during a migration from start to finish
		Amount of migrated data	Total quantity of data transferred in the process of migration
		Migration overhead	Additional resources utilized in a migration
3	Tafa et al. [3]	CPU consumption	Rate of usage of the processing resource
		Memory utilization	Rate at which the memory available is being used
		Total migration time	Total time required to move the VM between physical hosts
		Downtime	Time during which the service of the VM is not available
4	Akoush et al. [4]	Total migration time	Total time required to move the VM between physical hosts
		Migration downtime	Time during which the service of the VM is not available
		Migration link bandwidth	Inverse variation of total migration time and downtime
		VM memory size	Amount of memory available for VM
		Page dirty rate	Pace of modification of VM memory pages
		Pre- and post-migration overheads	Operations that are not part of the actual transfer process

(continued)

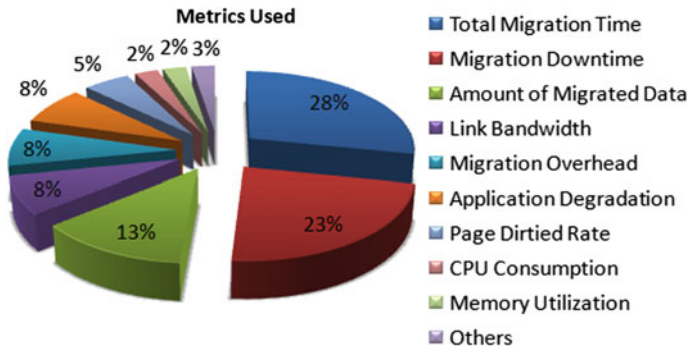
**Table 1** (continued)

Sl. no.	Author/s	Metric used	Metric description (if any)
5	Mohan and Shine [5]	Downtime	Time during which the service of the VM is not available
		Migration time	Total amount of time required to transfer a virtual machine at source node to destination node
6	Rakhi [6]	Total migration time	Overall time required to migrate a virtual machine
		Down time	Duration for which services are not available to the users
7	Kapil et al. [7]	Preparation time	Time at which VM's state transfer starts signalling start of the migration process
		Down time	Time duration for which the migrating VM is idle
		Resume time	Time between restart of VM execution at the target, indicating that the migration process is completed
		Application degradation	Magnitude by which migration slows down applications running on the VM
		Pages transferred	Overall number of memory pages transferred, throughout the entire time duration
		Total migration time	Overall amount of time from start to end
8	Soni and Kalra [8]	Total migration time	Time taken to migrate a virtual machine from current host to the target machine
		Down time	Time during which services are inaccessible by users
		Amount of migrated data	How much data is transferred form one host to another host

(continued)

**Table 1** (continued)

Sl. no.	Author/s	Metric used	Metric description (if any)
		Application degradation	Magnitude by which migration slows down applications running on the VM
		Migration overhead	System resource consumption
9	Hu et al. [9]	Total migration time	How long it takes to complete the migration from start to finish
		Down time	The time a VM is unresponsive during migration
10	Adami and Giordani [10]	Migration time	Lifetime of the TCP session
		Amount of transferred bytes	Amount of data exchanged by the 2 TCP sessions that are open during the VM migration process
11	Hines and Gopalan [11]	Preparation Time	Time between initiating migration and transferring the VMs processor state to the target node
		Resume time	Time between restart of VM execution at the target, signalling end of the migration process
		Pages transferred	Overall number of memory pages transferred, throughout the entire time duration
		Total migration time	This is the sum of all the above times from start to finish
		Application degradation	Magnitude by which migration slows down applications running on the VM



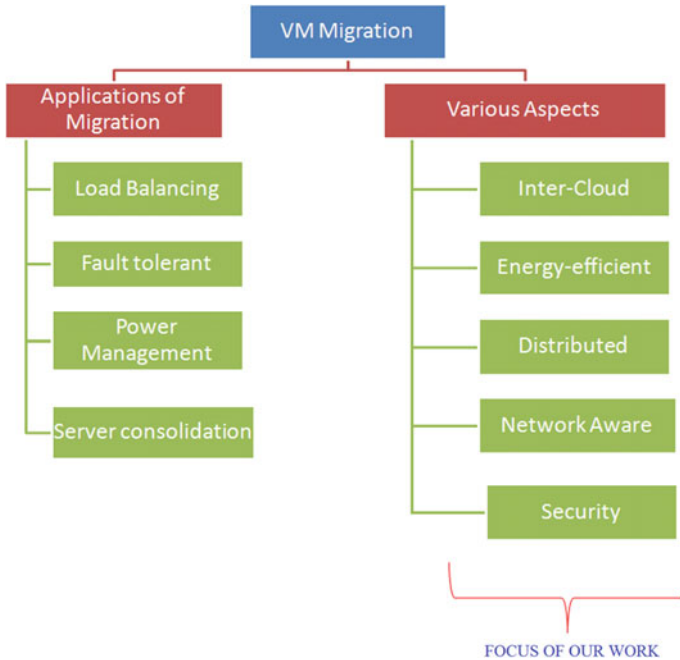
**Fig. 1** Survey of metrics used

### 3 A Survey on Various Aspects of Virtual Machine Migration

We can broadly classify the areas of research in virtual machine migration as shown in Fig. 2. At the top-level, they can be divided into two: applications of VM migration, or rather the reasons to perform migration. Another class includes the various aspects involved with VM migration such as the energy overhead, network performance degradation, and so on. As there are surveys of work in the first category, in this study, we shift our focus to the second category. In the following subsections, relevant works that discuss the different aspects of VM migration are discussed.

#### 3.1 Inter-Cloud VM Migration

While migrating among Clouds, migration may have to take across virtual machine monitors with different implementations. The authors propose a method that can efficiently migrate VMs across VMMs with different implementations in [12]. While using state-of-the-art methods to migrate VMs in such situations, we may encounter cases where the booting of the guest OS fails. To avoid this, a five-phased approach is used. The difference set of the files that are available at the source is calculated, and this difference set is transferred and overwritten at the destination. In order to address the security threat posed by VM migration in a Federated Cloud, an approach that makes use of remote attestation (RA) is proposed by Celesti et al. [13]. Suen et al. consider a different aspect of VM migration across Clouds. In this work, focus is given to the bandwidth and storage costs of data while migrating VMs [14]. One of the main difficulties while migrating VMs across Clouds is the requirement of



**Fig. 2** Classification of VM migration research

keeping the network state intact. The authors in [15] suggest a workaround for this problem that makes use of ViNe—a virtual networking approach at the user level which supports virtual overlay networks.

### 3.2 *Energy Efficient VM Migrations*

There are some methods by which the process of VM migration itself can be made energy efficient. For instance, the main aim of the authors in [16] is to conserve energy. For this, they consolidate VMs to lesser number of physical machines. Another work by Farahnakian et al., the main concern of the authors, is to lower the SLA violations and to reduce costs of power consumption. For this, the CPU utilization is approximated for a short time in future, based on its history [17]. Kord et al. have devised an algorithm to place VMs in a method that is energy efficient and also keep the SLA violations to a minimum [18]. The fuzzy analytic hierarchy process is applied to trade-off between these two requirements. In the work by Belglazov et al. [19], VM migration is used as a technique to consolidate VMs with an aim to reduce the number of working hosts at a time. The main focus of the authors in [20] is the measurement of incremental power consumption by VM migration.

### ***3.3 Distributed Approaches for VM Migration***

Heterogeneity is an inevitable factor in Cloud. A Cloud data center contains heterogeneous servers, and Virtual Machines have heterogeneous resource requirements depending on the type of applications they run. If proper allocation is not done, there can be numerous SLA violations and wastage of resources of the servers. The authors propose a policy-based agent mechanism in [21] for performing the above. Another team has proposed a distributed algorithm for virtual machine migration with the multiobjectives to reduce power consumption and also reduce the SLA violations to a minimum in [22]. The algorithm proposed makes use of two thresholds. The authors propose a distributed algorithm based on sampling that performs load balancing in [23]. One of the main objectives of this algorithm is zero-downtime for the VMs.

### ***3.4 Network-Aware VM Migration***

In some cases, there may be more than one migration going on at an instance. When multiple migrations like this take place, the network bandwidth may become a bottleneck. Thus, if proper scheduling is not employed, the migrations may take more time than is actually required [24]. In [25], the authors have proposed a migration approach that is network aware. They categorize VMs based on the type of workloads that they run, and this knowledge is then used to schedule migrations. To handle workloads that are dynamic in nature, migrations are generally used.

### ***3.5 Security***

Another important aspect that has to be taken care of while migrating VMs is the security. In [26], a model of the attacks on virtual systems is discussed and a security framework is proposed. In [27], data related to protection of the system is maintained in the hypervisors. While Cloud promises various advantages to the customers, the providers are also bound to ensure security. The ways in which an insider might pose security threats are explored in the work by Duncan et al. [28]. The insider can come in any form: the administrator of the Cloud provider, a staff member of the organization, or a person who makes use of the Cloud resources. In order to avoid mishandling of critical data when it lies within another territory, the authors in [29] propose a framework that can be used to detect live migration in the initial stages.

## 4 Research Issues

VM migration is an extensively researched area. According to the authors' understanding, the perceived issues related to research in VM migration are discussed in this section.

Comparatively, less work has been done in network-aware virtual machine migration in Cloud. There is a need for more Cloud simulators to provide virtual machine migration, so that the effects of virtual machine migration can be studied, and new issues arising due to VM migration can also be discovered.

The works done in the area of security and inter-Cloud mainly focus on Cloud providers at the IaaS level. There is a growing need to consider Cloud providers at the PaaS and SaaS level too. Comparatively, less works focus on the security aspect of VM migration. There should be more serious attempts to provide secured VM migration. Also, a migration technique to efficiently migrate VMs when there is no large bandwidth available is yet to be devised.

In grid and cluster, there exist methods that intelligently migrate VMs. Such an intelligent approach does not exist in Cloud. Such an intelligent decision model is yet to be introduced in Cloud.

## 5 Conclusion

VM Migration is one of the hot topics in Cloud computing. Here, we have discussed some of the research contributions to perform virtual machine migration, in general. There are many approaches for virtual machine migration, in grids, in Clouds, within datacenters, and so on. Here, a study from a different perspective on virtual machine migration has been done, considering the aspects of virtual machine migration that has not been the center of attention in previous works. The aspects include energy efficiency, network-aware approaches, and security. An attempt has been made to classify works in the literature based upon the aspect of virtual machine migration that they have considered. Some of the aspects which has been relatively less explored were also identified.

## References

1. Huang, W., et al.: High performance virtual machine migration with RDMA over modern interconnects. In: 2007 IEEE International Conference on Cluster Computing. IEEE (2007)
2. Huang, D., et al.: Virt-LM: a benchmark for live migration of virtual machine. ACM SIGSOFT Software Engineering Notes, vol. 36, no. 5. ACM (2011)
3. Tafa, I., et al.: The Comparison of Virtual Machine Migration Performance between XEN-HVM, XEN-PV and Open-VZ. ICT Innovations 2011. Springer Berlin Heidelberg 2012, pp. 379–394 (2011)

4. Akoush, S., et al.: Predicting the performance of virtual machine migration. In: 2010 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE (2010)
5. Mohan, A., Shine, S.: Survey on live VM migration techniques. *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)* (2013)
6. Rakhi, K.R.: Live virtual machine migration techniques—a survey. *Int. J. Eng. Res. Technol. (IJERT)* **1**(7) (2012)
7. Kamil, D., Emmanuel, S.P., Ramesh, C.J.: Live virtual machine migration techniques: survey and research challenges. In: 2013 IEEE 3rd International Conference on Advance Computing Conference (IACC) (2013)
8. Soni, G., Kalra, M.: Comparative study of live virtual machine migration techniques in cloud. *Int. J. Comput. Appl.* **84**(14) (2013)
9. Hu, W., et al.: A quantitative study of virtual machine live migration. In: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference. ACM (2013)
10. Adami, D., Giordani, S.: Virtual Machines Migration in a Cloud Data Center Scenario: An Experimental Analysis. *IEEE ICC*, pp. 2578–2582 (2013)
11. Hines, M.R., Gopalan, K.: Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. In: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual execution environments, New York, NY, USA, pp. 51–60 (2009)
12. Ashino, Y., Nakae, M.: VM migration method between different hypervisor implementations and its evaluation. In: 26th International Conference on Advanced Information Networking and Applications Workshops, pp. 1089–1094 (2012)
13. Celesti, A., Salici, A., Villari, M., Puliafito, A.: A remote attestation approach for a secure virtual machine migration in federated cloud environments. In: First International Symposium on Network Cloud Computing and Applications (2011)
14. Suen, C., Kirchberg, M., Lee, B.S., Suen, E.C., Lee, F.: Efficient migration of virtual machines between public and private cloud. In: Third IEEE International Conference on Cloud Computing Technology and Science, pp. 2–6 (2011)
15. Tsugawa, M., Riteau, P., Matsunaga, A., Fortes, J.: User-level virtual networking mechanisms to support virtual machine migration over multiple clouds. In: IEEE International Workshop on Management of Emerging Networks and Services, pp. 568–572 (2010)
16. Fang, S., Kanagavelu, R., Lee, B., Foh, C.H., Mi, K., Aung, M.: Power-efficient virtual machine placement and migration in data centers. In: IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (2013)
17. Farahnakian, F., Liljeberg, P., Plosila, J.: LiRCUP: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: 39th Euromicro Conference Series on Software Engineering and Advanced Applications, pp. 357–364 (2013)
18. Kord, N., Haghghi, H.: An energy-efficient approach for virtual machine placement in cloud based data centers. In: 5th Conference on Information and Knowledge Technology (IKT), pp. 44–49 (2013)
19. Beloglazov, A., Buyya, R.: Energy efficient allocation of virtual machines in cloud data centers. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 577–578 (2010)
20. Aikema, D., Mirtchovski, A., Kiddle, C., Simmonds, R.: Green cloud VM migration : power use analysis. *IEEE* (2012)
21. Gutierrez-Garcia, J.O., Ramirez-Nafarrate, A.: Policy-based agents for virtual machine migration in cloud data centers. In: IEEE International Conference on Services Computing (SCC) (2013)
22. Wang, X., et al.: A decentralized virtual machine migration approach of data centers for cloud computing. *Math. Problems Eng.* (2013)
23. Zhao, Y., Huang, W.: Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud. In: Fifth International Joint Conference on INC, IMS IDC, pp. 170–175 (2009)



24. Chen, H.: Network-aware coordination of virtual machine migrations in enterprise data centers and clouds. In: IFIP/IEEE International Symposium on Integrated Network Management (IM2013), vol. 1, pp. 888–891 (2013)
25. Stage, A., Setzer, T.: Network-aware migration control and scheduling of differentiated virtual machine workloads, pp. 9–14 (2009)
26. Shetty, J.: A Framework for Secure Live Migration of Virtual Machines, pp. 243–248 (2013)
27. Zhang, F., Chen, H.: Security-Preserving Live Migration of Virtual Machines in the Cloud. Springer Science+Business Media, LLC, pp. 562–587 (2013)
28. Duncan, A., Creese, S., Goldsmith, M., Quinton, J.S.: Cloud computing: insider attacks on virtual machines during migration. In: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications Cloud (2013)
29. Biedermann, S., Zittel, M., Katzenbeisser, S.: Improving security of virtual machines during live migrations. In: Eleventh Annual Conference on Privacy, Security and Trust (PST), pp. 352–357 (2013)

# Anomaly Detection in MapReduce Using Transformation Provenance



Anu Mary Chacko, Jayendra Sreekar Medicherla  
and S. D. Madhu Kumar

**Abstract** Data provenance is the metadata that captures information about data origin, how it was manipulated, and updated over time. Data provenance has great significance for big data applications as it provides mechanisms for verification of results. This paper discusses an approach to detect anomalies in Hadoop cluster/MapReduce job by reviewing the transformation provenance captured by mining the MapReduce logs. A rule-based framework is used to identify the patterns for extracting provenance information. The provenance information derived is converted into a provenance profile which is used for detecting anomalies in cluster and job execution.

**Keywords** Provenance • Transformation provenance • Big data  
Hadoop security • Anomaly detection

## 1 Introduction

Huge amounts of data are being produced by organizations every day. With the power of data analytics, conclusions and knowledge can be drawn out from the available raw data [1]. Provenance captures information on where the data came from, how it was derived, manipulated, combined, and how it has been updated over time [2]. Monitoring of provenance of a data item can provide circumstantial and contextual proof of its discovery and production. It provides mechanism for users to verify the data and therefore uses data confidently. Hence, there is a rising interest in evolving techniques to capture, store, and use provenance efficiently.

---

A. M. Chacko (✉) · J. S. Medicherla · S. D. Madhu Kumar  
National Institute of Technology, Calicut 673601, India  
e-mail: anu.chacko@nitc.ac.in

J. S. Medicherla  
e-mail: jayendrasreekar@gmail.com

S. D. Madhu Kumar  
e-mail: madhu@nitc.ac.in

Data provenance is broadly classified as *source provenance* and *transformation provenance*. *Source provenance* focuses on capturing the details of the source data items from which the current data item is derived. *Transformation provenance* focuses on capturing the details of the transformation process and captures the details of the process that led to the creation of current data item. Source refers to input and output of creation process, and transformation refers to the creation process itself. Provenance recording approaches are classified as *Lazy* and *Eager* based on when the provenance is generated. In *Lazy approach*, provenance is generated when it is requested, and in *Eager approach*, provenance is generated along with the creation of data. Data provenance is classified as *coarsely-grained* provenance and *fine-grained* provenance based on granularity of data considered [1].

Most of the schemes discussed in the literature focus on capturing source provenance for debugging purpose. Few works in eScience discuss capturing of transformation provenance for ensuring reproducibility of experiments [4].

Hadoop is open-source project that was designed to optimize handling of massive amount of data through parallelism using inexpensive commodity hardware. The earlier versions of Hadoop concentrated on task distribution, and very little attention was given to security. In later version, various techniques were provided like mutual authentication, enforcement of HDFS file permission, using tokens for authorization, etc. But Hadoop has a serious lack in detection of anomalous behavior.

In this paper, a novel approach for detecting anomalous behavior in Hadoop cluster/MapReduce job by capturing and querying transformation provenance of MapReduce workflow is discussed. The rest of the paper is organized as follows. Section 2 gives details of work related to the topic in the area of MapReduce workflows. Section 3 details the design of provenance capture system and Sect. 4 describes an approach for anomaly detection. Section 5 concludes the paper listing the future extensions possible.

## 2 Related Works

There are some existing works that capture provenance for MapReduce like RAMP [5], HadoopProv [6], Kepler + Hadoop [7], and LazyIncMapReduce [8]. The focus of these implementations is to capture data provenance to give explanation to the result obtained from MapReduce jobs. In this paper, the focus is to capture transformation provenance to provide improved security for MapReduce. SecureMR [9] is a work in this direction.

SecureMR [9] proposed by Juan Du et al. is a service integrity assurance framework for MapReduce. They present a decentralized replication-based integrity verification scheme to provide integrity for MapReduce in open systems. It consists of security components like secure task executor, secure verifier, secure committer,

secure scheduler, and secure manager. There is no scope for anomaly detection in this scheme.

Devarshi Ghoshal et al. proposed capture of provenance from log files [10]. In this work, the authors have proposed a model for collecting provenance data from logs and framework using rule-based identification and extraction. The advantage of this approach is that application logic is not disturbed.

In all approaches capturing source provenance for MapReduce workflow, the information is produced by modifying the application logic/framework to capture the required information during execution time. None of the schemes deal with transformation provenance. In this work, we have explored the option of capturing the transformation provenance through log files. Hadoop MapReduce logs are carefully observed for a particular job run to identify useful information. In this way, provenance is collected without modifying the application logic/framework. The information from the logs contributes to transformation provenance.

Hadoop does the data processing and scheduling in a way which is transparent to the user. There is a possibility that a compromised user or compromised node could do some malicious activity to gain additional resource usage and obstruct services to the other nodes for its purposes. An attacker could perform some attacks to slow down the data processing and create a denial-of service situation in the cluster. Currently, any such anomalous activity would go unnoticed despite having security features enabled in Hadoop. Transformation provenance captured can throw light on these malicious activities.

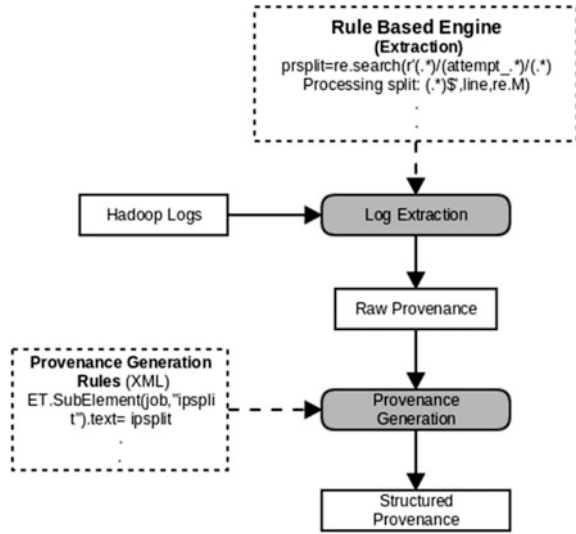
### 3 Design of Provenance Profile

A rule-based engine is used to mine the logs for capturing provenance. Once a job is run on Hadoop MapReduce cluster, the log information is generated across the cluster in different nodes by various Hadoop daemon services like *NameNode*, *DataNode*, *TaskTracker*. This information is distributed among all the nodes. SSH is used to aggregate all the log files to a common location. Once the log information is aggregated from all the slave nodes; the required information is extracted from the log files using the set of rules. The extracted information is processed and stored in XML to create the provenance profile. The profile generated is used for anomaly detection.

Rule-based provenance generation has three steps. First step is to identify the required patterns in the logs and create rules. Second step is to extract the required information from the logs using rules. In the third and final step, the provenance is deduced from the extracted information. The steps in rule-based generation using proposed provenance generation framework are shown in Fig. 1.

A Hadoop job typically has many Map and Reduce tasks. All the Map and Reduce tasks in the job follow the same execution pattern. A pattern can be observed even in the Read/Write, cluster setup operations. Due to these patterns, the

**Fig. 1** Rule-based provenance generation framework



log information for all the tasks has definite pattern. These patterns in the log files are identified, and rules are created so as to extract them. The main advantage with the rule-based extraction from the logs is that rules can be updated/changed/removed/added based on the requirement without modifying application logic.

Hadoop generates detailed log information of all the services running in the cluster like *NameNode*, *DataNode*, *JobTracker*, *TaskTracker*. Hadoop uses *Log4j framework* for generating log information with levels like *WARN*, *INFO*, *ERROR*. Additional logs can be generated by extending *Log4j* in our application logic.

A Hadoop cluster with three machines, one master node “hmaster” and two slave nodes “sc1” and “sc2” was set up. The “word count” problem was run on input datasets of varying range. The logs that are generated are observed.

Python regular expression was used for matching the required text and to extract the required information using rules. Provenance profile is constructed using the extracted information. As the provenance profile is to be used for anomaly detection and debugging, provenance needs to be stored in a structured manner so that it can be easily queried. Hence, XML format is used for storing the provenance information. For every job execution, provenance profile is created with unique identifier. The provenance profile contains the complete information about the execution of the job run, cluster configuration information as well as *ERROR* and *WARN* messages which can be used to debug any errors or improper configuration setup in the Hadoop cluster. Figures 2 and 3 show snapshots of a provenance profile that is generated.

In the next section, the use of the provenance profile for anomaly detection is explained.

```

- <prov-prof jobid="job_201604281216_0001">
- <jobconfig>
  <jobid>job_201604281216_0001</jobid>
  <user>hduser</user>
  - <clusterconf>
    <name>hmaster</name>
    <ip>192.168.4.178</ip>
    <name>sc2</name>
    <ip>192.168.4.76</ip>
    <name>sc1</name>
    <ip>192.168.4.108</ip>
    <tmpdir>/app/hadoop/tmp</tmpdir>
    <opdir>/user/hduser/ouput</opdir>
    <ipdir>hdfs://hmaster:54310/user/hduser/input</ipdir>
    <dfsperm>755</dfsperm>
    <dfsblocksize>67108864</dfsblocksize>
    <nmap>10</nmap>
    <nred>1</nred>
  </clusterconf>
</jobconfig>
- <jobinfo>
  . . .

```

Fig. 2 Cluster information

```

- <job>
  <job-type>MAP</job-type>
  <attempt-id>attempt_201604281216_0001_m_000000_0</attempt-id>
  <task-id>task_201604281216_0001_m_000000</task-id>
  <location>tracker_sc2:localhost/127.0.0.1:41727</location>
  - <ipsplit>
    hdfs://hmaster:54310/user/hduser/input/big2.txt:0+67108864
  </ipsplit>
  <op-size>1251830</op-size>
  <jvm-id>jvm_201604281216_0001_m_1407151495</jvm-id>
  <status>completed</status>
  <time>23697</time>
  - <RWop>
    <src>192.168.4.76:50010</src>
    <dest>192.168.4.76:35807</dest>
    <bytes>1139</bytes>
    <op>HDFS_READ</op>
  - <cliID>
    DFSClient_attempt_201604281216_0001_m_000000_0_-1681988325_1
  </cliID>
  <offset>0</offset>
  <srvID>DS-1146571055-192.168.4.76-50010-1461514652516</srvID>
  <blockid>blk_-5926159342225058141_1115</blockid>
  <duration>255311</duration>
</RWop>
- <RWop>
  <src>192.168.4.76:50010</src>
  <dest>192.168.4.76:35810</dest>

```

Fig. 3 Job run information

## 4 Anomaly Detection Using Provenance Profile

There are many attacks possible from external as well as internal entities in a Hadoop system. A malicious user or malicious program submitted by the user or improper configuration of a cluster can lead to information leakage.

With the provenance profile generated, certain anomalous behavior of job run or anomalous behavior of the cluster can be detected. A set of checks is performed to make sure that there is no activity which is deviant from the normal. In addition to that, provenance profile provides information useful for debugging.

Anomalies like invalid task inputs or data leakage due to output getting stored in improper location can be detected using the profile created. In addition to this, using the profile, the number of tasks performed and their running time and status of nodes in the cluster can be validated.

In order to understand the anomaly detection using provenance profile, the following attack scenarios were simulated and the results of the various checks are discussed below.

### A. Check input and output folders of tasks.

Every Map task takes the input from an HDFS location, and Reduce tasks write output to HDFS location. There can be a possibility that the output is stored in a location which is a public folder set by improper configuration or malicious user. This puts the confidential information at risk. Figure 4 shows an instance where a user tries to give input directories which are not valid. A mismatch between input folder path submitted and input split path raises an anomaly.

```
Check-1 : Input to all the tasks are valid inputs.

ERROR : (ANAMALOUS ACTIVITY) Task is taking input from an invalid directory -----
-----
Check-2 : Output are stored in proper locations

INFO: Task is storing output to the correct directories
-----
Check-3 : Checking Number of tasks performed

There are 14 map attempts and 11 map tasks and 1 reduce tasks
-----
Check-4 : Status of nodes in cluster

INFO : No Anamalous Behaviour from the nodes in the cluster. ['hmaster', 'sc1', 'sc2'] nodes performed [2,
6, 6] completed Tasks respectively
-----
Check-5 : Analyzing Task Execution times

Mean Values of Map and Reduce Exection times are : 50639.0909091 257207.0
Standard Deviation of Map and reduce Execution times are 40845.421073 0.0
There is a deviation of 80.6598624495 % of values from mean for Map task and 0.0 % of values from mean for r
educer task
```

Fig. 4 Anomaly detection improper input

### B. Checking total number of tasks executed.

As soon as the job is submitted to the Hadoop cluster, it is divided into many tasks based on the number of splits. The total number of splits and number of reducers from the job configuration file can be verified with the tasks information from the provenance profile to identify whether any computations that were skipped.

### C. Checking status of nodes in the cluster.

Hadoop runs on many nodes in a cluster. Users are not aware of what is running in the cluster. Even though some of the nodes are not properly configured or not have started properly for some reason, the cluster will run with the available slaves in the cluster. Here, there is an underutilization of resources happening. Using provenance profile, the status of the nodes in the cluster can be checked to identify any inactive node in the cluster. Figures 5 and 6 show scenarios where cluster is not completely utilized and one or more slave node skipping computations.

### D. Analyzing task run time

All Map and Reducer tasks perform similar execution patterns. If the input size is same to all the MAP tasks, they should perform similarly. Hence, there should be a similarity in the execution times of MAP tasks as well. SYN flooding attack was performed on one of the slave machines in a cluster of three machines to make the slave system less responsive.

The run times of all the map tasks were collected with and without attack. The mean and standard deviation for both the set of values were calculated. It was found

```
Check-1 : Input to all the tasks are valid inputs.

INFO: Tasks taking input from the correct directories
-----
Check-2 : Output are stored in proper locations

INFO: Task is storing output to the correct directories
-----
Check-3 : Checking Number of tasks performed

There are 10 map attempts and 10 map tasks and 2 reduce tasks
-----
Check-4 : Status of nodes in cluster

ERROR : (ANAMALOUS ACTIVITY) The Node scl in the cluster Seems to be inactive. Possibility of Attack or improper configuration of the cluster
Go through debug in provenance xml

tasks performed by nodes are [0, 7, 7]
-----
Check-5 : Analyzing Task Execution times

Mean Values of Map and Reduce Exection times are : 27318.3 26904.0
```

Fig. 5 Anomaly detection inactive slave



```

Check-1 : Input to all the tasks are valid inputs.

INFO: Tasks taking input from the correct directories
-----
Check-2 : Output are stored in proper locations

INFO: Task is storing output to the correct directories
-----
Check-3 : Checking Number of tasks performed

There are 10 map attempts and 10 map tasks and 1 reduce tasks
-----
Check-4 : Status of nodes in cluster

ERROR :(ANAMALOUS ACTIVITY) The Node sc1 in the cluster Seems to be inactive. Possibility of Attack or imp
roper configuration of the cluster
Go through debug in provenance xml
ERROR :(ANAMALOUS ACTIVITY) The Node sc2 in the cluster Seems to be inactive. Possibility of Attack or imp
roper configuration of the cluster
Go through debug in provenance xml

tasks performed by nodes are [13, 0, 0]
-----

```

Fig. 6 Anomaly detection inactive slave nodes

that when there is no attack, the standard deviation value was very less. In the other case, when there is an attack, the deviation is high (approx 50%) from the mean which indicates that the run times of map tasks are varying with a high value. This information captured in provenance profile can indicate possible attack. Figure 7 shows anomalies of job in a cluster under flooding attack.

```

Check-1 : Input to all the tasks are valid inputs.

INFO: Tasks taking input from the correct directories
-----
Check-2 : Output are stored in proper locations

INFO: Task is storing output to the correct directories
-----
Check-3 : Checking Number of tasks performed

There are 10 map attempts and 10 map tasks and 1 reduce tasks
-----
Check-4 : Status of nodes in cluster

INFO : No Anomalous Behaviour from the nodes in the cluster. ['hmaster', 'sc1', 'sc2'] nodes performed [4,
5, 4] completed Tasks respectively
-----
Check-5 : Analyzing Task Execution times

Mean Values of Map and Reduce Exection times are : 113192.7 323908.0
Standard Deviation of Map and reduce Execution times are 89495.8571623 0.0
There is a deviation of 79.0650432071 % of values from mean for Map task and 0.0 % of values from mean for r
educe task

```

Fig. 7 Anomaly detection for detecting possible SYN attack

## 5 Conclusion and Future Work

This paper demonstrates a rule-based approach to generate transformation provenance from MapReduce log files in Hadoop ecosystem. The provenance captured is used to build a provenance profile for the job. The use of provenance profile to detect anomalous behavior of the job/cluster was demonstrated. The advantage of this approach is that it doesn't involve any modification to MapReduce framework/application. As it is run as a background process, provenance collection does not cause any computational overhead for jobs during execution. This work can be extended to explore how provenance profiles are useful to counter-advanced attacks like man-in-the-middle attacks and Replay attacks. It will also be an interesting work to enhance the profile by collecting more log files from other Hadoop ecosystem components like Pig, Zookeeper, and HBase.

## References

1. Glavic, B., Dittrich, K.: Data provenance: a categorization of existing approaches. In: Proceedings of the 12th GI Conference on Datenbanksysteme in Business, Technologie und Web (2007)
2. Ikeda, R., Widom, J.: Panda: a system for provenance and data. *IEEE Data Eng. Bull. Spec. Issue Data Provenance* **33**(3), 42–49 (2010)
3. Rama, S., Liu, J.: Understanding the semantics of data provenance to support active conceptual modeling. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4512, pp. 17–29. LNCS (2008)
4. Simmhan, Y.L., Pale, B., Gannon, D.: A survey of data provenance in e-science. *SIGMOD Rec.* **34**(3), 31–36 (2005). <https://doi.org/10.1145/1084805.1084812>
5. Ikeda, R., Widom, J.: Ramp: a system for capturing and tracing provenance in map reduce workflows. In: *International Conference on Very Large Databases (August 2011)*
6. Akoush, S., Sohan, R., Hopper, A.: Hadoopprov: towards provenance as a first class citizen in mapreduce. In: Presented as part of the 5th USENIX Workshop on the Theory and Practice of Provenance. USENIX, Berkeley, CA (2013)
7. Crawl, D., Wang, J., Altintas, I.: Provenance for mapreduce-based data-intensive workflows. In: *Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science WORKS'11*, pp. 21–30 (2011)
8. Wei, W., Du, J., Yu, T., Gu, X.: Securemr: a service integrity assurance framework for mapreduce. In: 2009 Annual Computer Security Applications Conference, ACSAC'09, pp. 73–82 (2009)
9. Ghoshal, D., Plale, B.: Provenance from log files: a bigdata problem. In: *ACM International Conference Proceeding Series*, pp. 290–297 (2013)
10. Chacko, A., Madhu, S., Madhu Kumar S.D., Gupta, A.: Improving execution speed of incremental runs of mapreduce using provenance. In: *Special Issue on Big Data Visualization and Analytics. Inderscience Publishers (In Press)* (2016)

# Evaluation of MapReduce-Based Distributed Parallel Machine Learning Algorithms



Ashish Kumar Gupta, Prashant Varshney, Abhishek Kumar, Bakshi Rohit Prasad and Sonali Agarwal

**Abstract** We are moving toward the multicore era. But still there is no good programming framework for these architectures, and therefore no general and common way for machine learning to take advantage of the speedup. In this paper, we will give framework that can be used for parallel programming method and that can be easily applied to machine learning algorithms. This work is different from methods that try to parallelize an individual algorithm differently. For achieving parallel speedup on machine learning algorithms, we use MapReduce framework. Our experiments will show speedup with an increasing number of nodes present in cluster.

**Keywords** MapReduce · Naïve Bayes · K-means · Linear regression  
Hadoop

## 1 Introduction

Machine learning deals with the set of topics related to the creation and evaluation of algorithms that facilitate classification, pattern recognition, and prediction based on models derived from existing data. Machine learning algorithms helped in automate the tasks which are impossible doing manually by humans. But the

---

A. K. Gupta (✉) · P. Varshney · A. Kumar · B. R. Prasad · S. Agarwal  
Indian Institute of Information Technology, Allahabad, India  
e-mail: iit2013020@iiita.ac.in

P. Varshney  
e-mail: iit2013084@iiita.ac.in

A. Kumar  
e-mail: iit2013099@iiita.ac.in

B. R. Prasad  
e-mail: rs151@iiita.ac.in

S. Agarwal  
e-mail: sonali@iiita.ac.in

amount of data is increasing every day enormously. It has become tough even for machines to analysis this much of data. There is a need of finding methods to tackle the problem without compromising the accuracy of these algorithms. MapReduce is such a programming model for processing large data sets. Users need to specify a map function that takes a key-value pair as an input and output an intermediate key-value, and then, reducer functions merges all the values of each key. The run-time system handles the details of partitioning the input data, then scheduling execution of program on different machines present in the cluster, machine failures, manage inter-machine communication. This allows programmers to easily utilize the resources of large distributed systems. Many real-world tasks are expressible in this model, but transforming machine learning algorithms to fit in MapReduce framework is a bit difficult. In this paper, we will work on three such algorithms; K-means, Naïve Bayes, and linear regression that fit into this framework. Then, we will analyze the effect of increasing nodes in cluster on the speedup of these algorithms.

## 2 Related Work

Production of data is expanding at an extreme rate. Experts now point to a 4300% increase in annual data generation by 2020. In 2012, people start storing about 1 EB of data which reached to 7.9 ZB in 2015 and will reach to 35 ZB 2020. These are the few facts which tell us how fast the digital data is increasing. This data is generally unstructured or semistructured. Not all data is important but the whole data needs to be analyzed to get useful information. On the same hand, we are in the era where number of cores per chip is increasing. But even then there is not a good framework for machine learning. There are many programming languages for parallelization such as ORCA, Occam ABCL, SNOW but does not give a general and obvious method for parallelization of algorithms. The vast literature on data mining and learning in parallel computing [1] but it does not give a general framework for programming machine learning on multicore. These literatures contain ingenious, long and distinguished ways to speed up individual learning algorithms, for example, cascaded SVMs [2]. But these do not yield general techniques for parallelization, thus lack in widespread use of them. General machine learning approach given by Jin and Agrawal [3] but only for machines with shared memory. Caregea et al. [4] is an example of general papers which give data distribution conditions for parallel machine learning, but it was limited to the decision trees. Many researchers have given many parallel clustering algorithms [5–7]. But these algorithms have following drawbacks: first they assume that all objects can reside in main memory at the same time and second is that their systems used for parallelization provided restricted programming models and then used those restrictions to parallelize the computation. But the techniques were not applicable for large datasets. MapReduce framework [8–11] is a programming model that is developed for processing large data sets that can be applied to many real-world

tasks. Users need to only write mappers and reducers framework automatically parallelize the execution and handle the data transfer and communication between nodes by itself. Google and Hadoop provide this framework with automatically handling dynamic flexibility support [12] and fault tolerance.

### 3 Proposed Methodology

Proposed work deals with three significant algorithms of machine learning, i.e., Naïve Bayes, K-means, and linear regression. Further subsections will specify their working in MapReduce setting in order to achieve scalability on top distributed parallel environment of Hadoop.

#### 3.1 Naïve Bayes

Naive Bayes models assign class labels to data instance, represented as vectors of features. Naive Bayes classification algorithm is the easiest and simple supervised machine learning algorithm which is based on the Bayes’ theorem as follows:

$$P(A|B) = P(B|A) *P(A) / P(A)$$

MapReduce model is built to estimate the probabilities as shown depicted in Fig. 1.

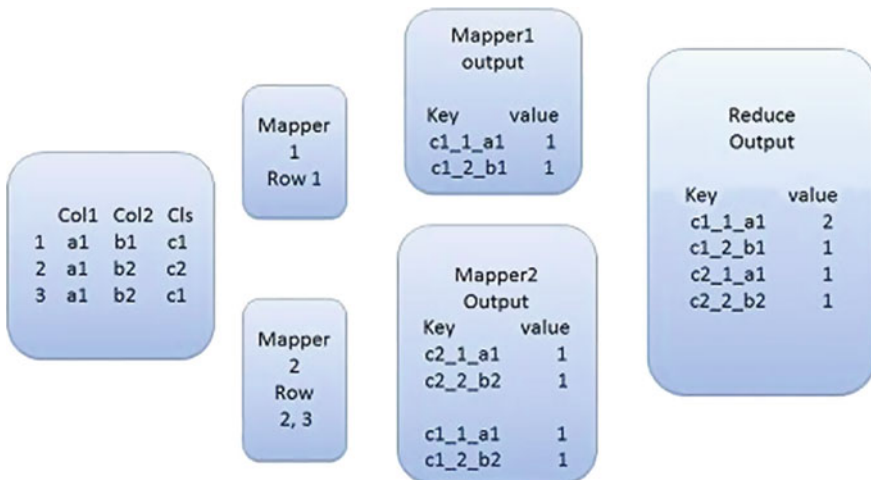


Fig. 1 Naïve Bayes training process in MapReduce environment

With the help of MapReduce, distributed nature of model will be able to handle large data set. Given data instance  $d$  and class  $c$ , to find the class which maximizes  $P(c|d)$ , we need to find  $P(d|c) * P(c)$ .  $d$  will be represented as the collection of attributes in a vector  $d = \{x_1, x_2, \dots, x_n\}$ . Therefore, we need to find  $P(c|d) = P(x_1|c) * P(x_2|c) * \dots * P(x_n|c) * P(c)$  for each class at the time of testing. Thus, if we know count of  $x_i$  in class  $c_j$  and count of  $c_j$ , we can easily give answer in real time. To count these values, we use mappers, combiners, and reducers as discussed below:

**Mappers:** Take input instance by instance from input file. Then output key-value pair where key will be class-value\_attribute-index\_attribute-value and value will be simply an integer 1.

**Combiners:** Take the output of a single mapper and sum the values of a key and output the key and sum value.

**Reducers:** Take the input of different combiners. It gets all the values of a key from various combiners. It sums up the value of a key write in a common file key and its count in the whole input.

### 3.2 *K-means*

Clustering is used to cluster all the similar data instances in a large set of data. That is the reason it is also called data segmentation. It partitions the large data set into smaller clusters. It is unsupervised learning because with each data instance class label is absent. Suppose a data set  $D$  has  $n$  finite objects in Euclidean space. Then if clustering is applied on  $D$ , it will make  $K$  clusters  $C_1, C_2, \dots, C_k$  that is each  $C_i$  is a subset of the data set  $D$  or  $C_i \subset D$  and  $C_i \cap C_j = \emptyset$  for  $(1 \leq i, j \leq k)$ . Difference between two point  $p \in C_i$  and  $C_i$  (centroid) is measured by the Euclidean distance between them and represented by  $dist.(p, C_i)$ . MapReduce-based model of  $K$ -means clustering includes following mapper, reducer, and combiner operations.

**Mappers:** Read data set instance by instance. Each mapper will also be given a common file having  $k$ -centers initially which will be random. Now mapper will find the nearest center to the instance and output the key-value pair where key will be cluster number and value will be the whole instance.

**Combiners:** If reducers receive data from mappers directly means all the values for a key are received by a single reducer, the amount of data could become very large; thus, a task could become heavy for the reducers. Thus, combiner reads output emitted by the single mapper and sums the respective attribute values of the instances emitted as value from mappers of a key (cluster), Combiners output cluster index as key and sum and count of instances per cluster as value.

**Reducers:** Reducers, add all the values of keys emitted by all combiners, for each cluster we have the sum of instances which makes the cluster and its count as well, and then we can get mean of the cluster by dividing sum by count. This will give new cluster centers. This completes one iteration. After each iteration reducers generate new cluster centers these will be used to feed mapper. The clustering process is diagrammatically represented by Fig. 2.

### 3.3 Linear Regression

Linear regression is a method to fit a curve such that it minimizes the error between the actual data points and the point predicted by the curve. A relationship is established between output variable Y and one or more independent variable X.

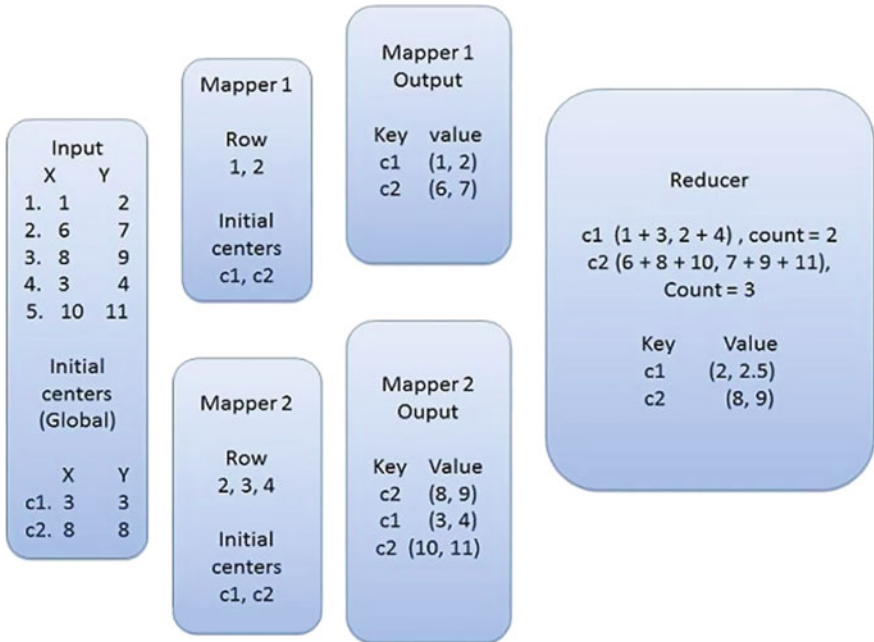


Fig. 2 K-means iteration in MapReduce environment

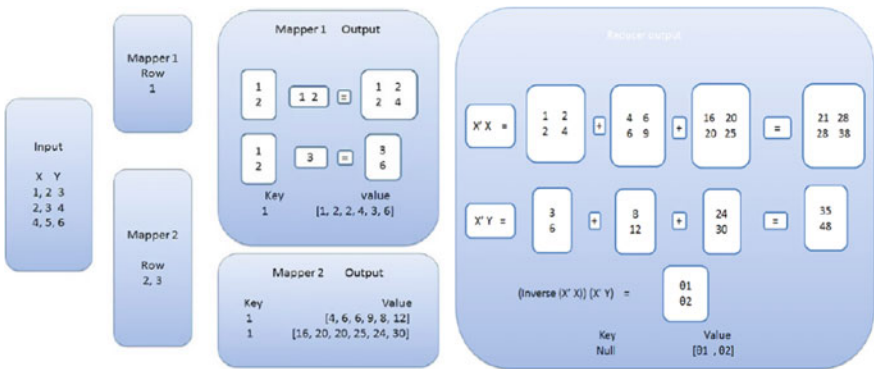


Fig. 3 Linear regressor in MapReduce environment

$$Y = x_1 + x_2 + x_3 + x_4 + \dots + x_n$$

In least squares (linear regression), which fits  $y = \theta'x$  by solving:  $\theta^* = \min_{\theta} (\theta'x_i - y_i)^2$ . Parameter  $\theta$  is typically solved for by matrix  $X \in R^{m \times n}$  whose rows are training instances  $X_1, X_2, X_3, \dots, X_m$  instances and  $\sim Y = [Y_1, Y_2, \dots, Y_m]$  be the vector of target labels and solving the normal equation to obtain  $\theta^* = (\text{Inverse}(X' X)) X' \sim Y$ . In MapReduce model, we need to compute  $A = X' X$  and  $B = X' \sim y$  as follows:

**Table 1** Dataset description

Dataset	#Attributes	#Instances	Remarks
MUSHROOM	23	8416	Concatenated the data file to make it 3.3, 4.5 and 6.6 GB
BIKE SHARING	12	17389	Concatenated the original dataset to make it 1, 1.5 and 2 GB.
INPUT.TXT	2	900000000	Each instance is a random point.

**Table 2** System configuration for experiments

Configuration of each machine	Parameter value
Machine used	Intel Core-i2 CPU
Processor Speed	2.8 GHz
RAM	4 GB
Number of Cores	4
Operating System	64-bit Linux (Ubuntu 12.04)
Hadoop Version	2.7.3
JDK Version	1.8.0_111

**Table 3** Evaluation of distributed Naive Bayes algorithm

Data size (GB)	#Nodes	Accuracy (%)	#Mappers	#Reducers	Time (min)
3.3	2	87.4	25	1	9.5
3.3	3	87.4	25	1	4.5
3.3	4	87.4	25	1	3
4.5	2	87.4	34	1	13
4.5	3	87.4	34	1	6.2
4.5	4	87.4	34	1	4
6.6	2	87.4	39	1	19.2
6.6	3	87.4	39	1	9.2
6.6	4	87.4	39	1	5.9

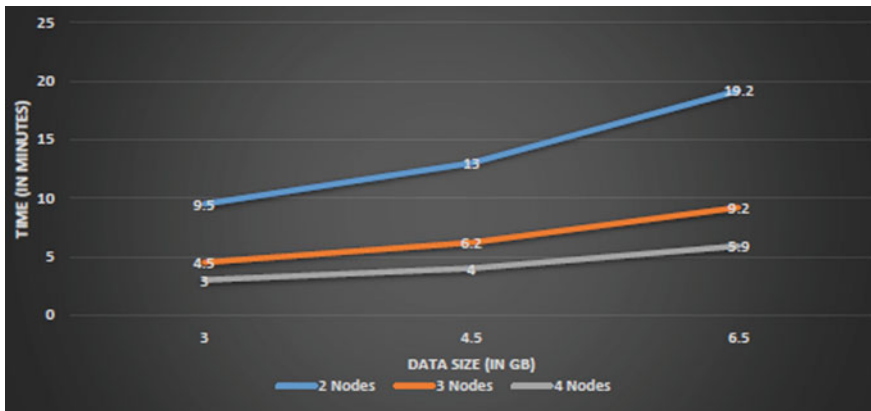


**Table 4** Evaluation of distributed K-means algorithm

Data size (GB)	#Nodes	#Mappers	#Reducers	Time (min)
3.3	2	1.6	2	12
3.3	3	1.6	3	12
3.3	4	1.6	4	12
4.5	2	2.4	2	18
4.5	3	2.4	3	18
4.5	4	2.4	4	18
6.6	2	3.2	2	24
6.6	3	3.2	3	24
6.6	4	3.2	4	24

**Table 5** Evaluation of distributed linear regression algorithm

Data size (GB)	#Nodes	RMSE	#Mappers	#Reducers	Time (min)
1	2	10.63	8	1	91.4
1	3	10.63	8	1	36.75
1	4	10.63	8	1	30.25
1.5	2	10.63	12	1	110.25
1.5	3	10.63	12	1	68.5
1.5	4	10.63	12	1	54.1
2	2	10.63	25	1	212
2	3	10.63	25	1	88
d2	4	10.63	25	1	55.75



**Fig. 4** Data size versus time plot

$$A = \sum_{i=1}^m (X'_i X_i) \text{ and } B = \sum_{i=1}^m X'_i Y_i.$$

A and B can be divided into equal size pieces and distributed among nodes. The process is exhibited in Fig. 3. Following mapper and reducer operations takes place.

**Mapper:** Mapper takes a row transpose (column vector of  $N \times 1$  dimension) and multiple it by the same row ( $N \times 1$ ) to get  $N \times N$  matrix. Also, it multiply row transpose with the output value of this row  $y$ . key will be any fix value for all mappers (let say 1) and output will be matrix and this matrix will be flattened in the form of string.

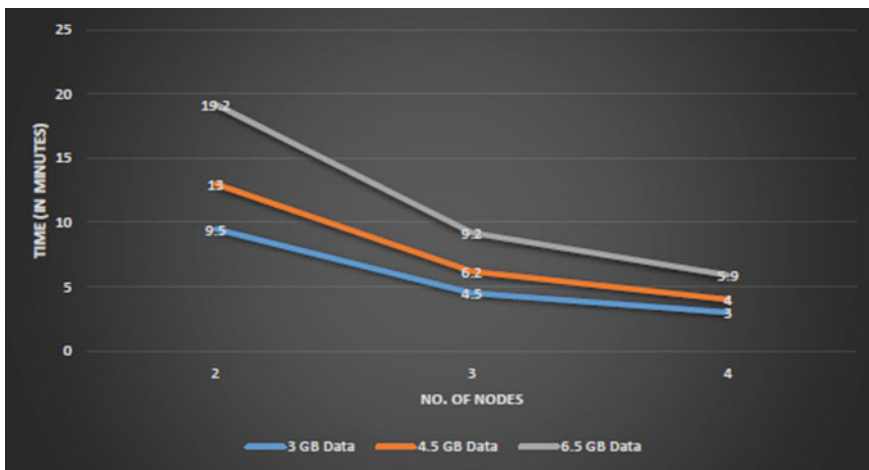


Fig. 5 Node versus time plot

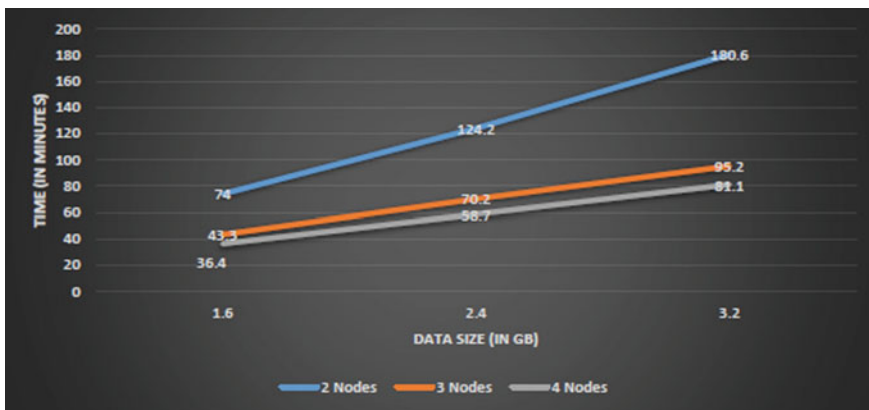


Fig. 6 Data size versus time plot

**Reducer:** Reducer takes each input string and recreates matrix and vector from it.

Add all matrixes and vectors separately. Finally, perform a matrix inverse and multiple it by vector to get  $\theta$  vector of length N.

### 3.4 Dataset Description and Experimental Settings

Naive Bayes and linear regression algorithms use MUSHROOM dataset and BIKE SHARING dataset, respectively, taken from UCI repository for machine

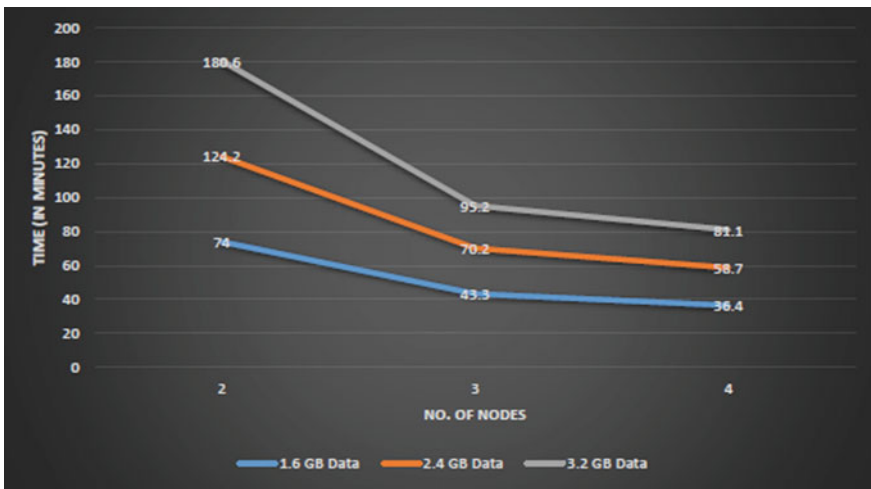


Fig. 7 Node versus time plot

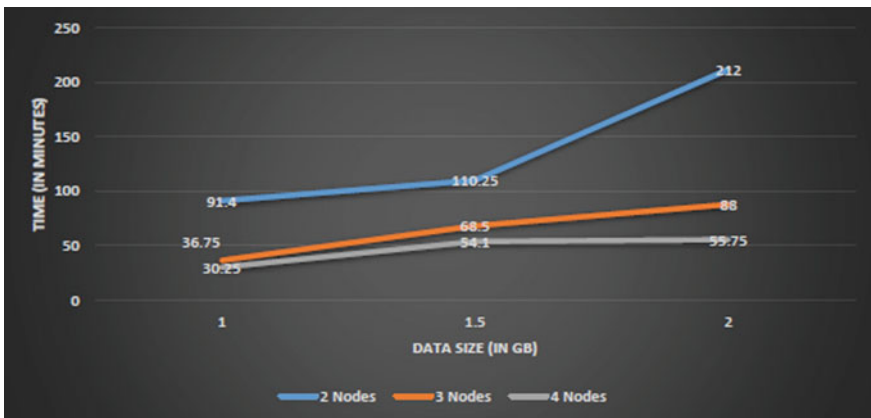


Fig. 8 Data size versus time plot

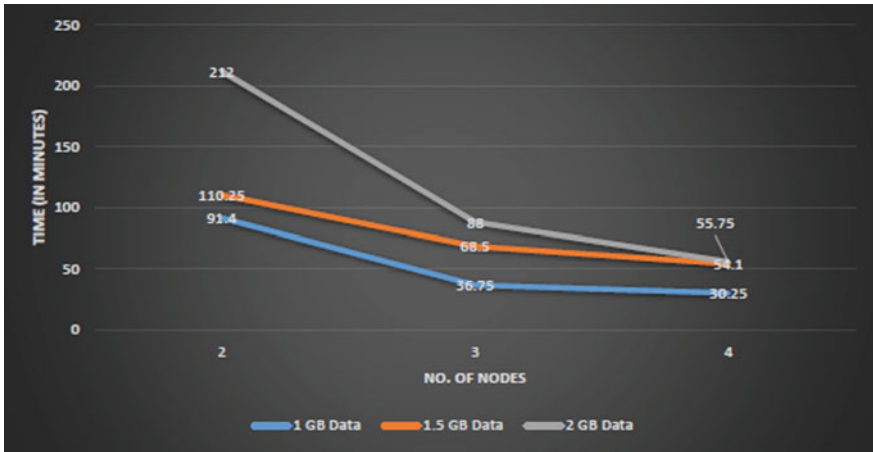


Fig. 9 Node versus time plot

learning. However, K-means algorithm uses dataset in a input.txt file which contains randomly generated x and y coordinates. Description of datasets is given in Table 1. Experiments are performed with the system settings as mentioned in Table 2.

## 4 Results and Discussion

After extensive run of experiments for each algorithm with different number of nodes and varying data sizes Tables 3, 4, and 5 are obtained which shows relevant details of findings. Moreover, Figs. 4, 5, 6, 7, 8, and 9 specify the scalability and speedup gains obtained by these algorithms in MapReduce computing model on Hadoop framework.

## 5 Conclusion

In time versus nodes graph, we can observe if we double the data size the time drop in 2 nodes is more than half but approximately half in case of 3 and 4 nodes. In time versus node graph, we can see if we double the nodes for particular data size time drop for Naïve Bayes is  $(1/3)$ ,  $(1/2)$  for K-means but less regular effect in case of linear regression.

## References

1. Liu, K. et al.: Distributed data mining bibliography (2006)
2. Graf, H.P. et al.: Parallel support vector machines: the cascade svm. *Adv. Neural Inf. Process. Syst.* (2004)
3. Jin, R., Yang, G., Agrawal, G.: Shared memory parallelization of data mining algorithms: techniques, programming interface, and performance. *IEEE Trans. Knowl. Data Eng.* **17**(1), 71–89 (2005)
4. Caragea, D., Silvescu A., Honavar V.: A framework for learning from distributed data using sufficient statistics and its application to learning decision trees. *Int. J. Hybrid Intell. Syst.* **1** (1–2), 80–89 (2004)
5. Rasmussen, E.M., Willett, P.: Efficiency of hierarchic agglomerative clustering using the ICL distributed array processor. *J. Doc.* **45**(1), 1–24 (1989)
6. Li, X., Fang, Z.: Parallel clustering algorithms. *Parallel Comput.* **11**(3), 275–290 (1989)
7. Olson, C.F.: Parallel algorithms for hierarchical clustering. *Parallel Comput.* **21**(8), 1313–1325 (1995)
8. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. *Commun. ACM* **51**(1), 107–113 (2008)
9. Ranger, C. et al.: Evaluating mapreduce for multi-core and multiprocessor systems. In: 2007 IEEE 13th International Symposium on High Performance Computer Architecture. IEEE (2007)
10. Lämmel, R.: Google’s mapreduce programming model—revisited. *Sci. Comput. Program.* **70** (1), 1–30 (2008)
11. Shvachko, K. et al.: The hadoop distributed file system. In: 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE (2010)
12. Ghemawat, S., Gobiuff, H., Leung, S.-T.: The Google file system. In: *ACM SIGOPS Operating Systems Review* vol. 37, no. 5. ACM (2003)
13. Borthakur, D.: The hadoop distributed file system: Architecture and design. *Hadoop Project Website* **11**(2007), 21 (2007)
14. Liu, X. et al.: Implementing WebGIS on hadoop: a case study of improving small file I/O performance on HDFS. In: 2009 IEEE International Conference on Cluster Computing and Workshops. IEEE (2009)
15. Mohandas, N., Thampi, S.M.: Improving hadoop performance in handling small files. In: *International Conference on Advances in Computing and Communications*. Springer, Berlin, Heidelberg (2011)

# TSED: Top-k Ranked Searchable Encryption for Secure Cloud Data Storage



B. Lydia Elizabeth, A. John Prakash and V. Rhymend Uthariaraj

**Abstract** With the proliferation of data and appealing features of cloud storage, a large amount of data are outsourced to the public cloud. To ensure privacy of sensitive data and defend against unauthorized access, data owners encrypt the data. Therefore, it is essential to build efficient and secure searchable encryption techniques. In this paper, we address two problems. Firstly, most of the existing work uses inverted index for faster retrieval; however, inverted index is not inherently dynamic. Secondly, due to a large amount of data and on-demand users, secure and efficient ranked retrieval sorted by relevance is desired especially in the pay-as-you-use cloud model. In this paper, we construct TSED: top-k ranked searchable encryption for secure cloud data storage. TSED uses a dynamic and secure index using homomorphic encryption and efficient ranked retrieval. Performance analysis on real-world dataset shows that TSED is efficient and practical.

**Keywords** Cloud security · Privacy · Top-k ranked search · Confidentiality

## 1 Introduction

In recent years, due to the exponential growth of data, the popularity of storage has gained considerable momentum and has a great promise for growth. Individuals and enterprises rely more on digital data, and cloud storage has become an integral part of the modern mobile world. The benefits of storing the data in the public infrastructure cloud are undeniable. However, security is an important barrier for adopting the cloud for both the large enterprises and small and medium enterprises

---

B. L. Elizabeth · A. J. Prakash (✉) · V. R. Uthariaraj  
Anna University Chennai, Chennai, Tamilnadu, India  
e-mail: johnprakash@annauniv.edu

B. L. Elizabeth  
e-mail: lydiajohn@annauniv.edu

V. R. Uthariaraj  
e-mail: rhymend@annauniv.edu

(SMEs) [1]. The working principle of public cloud storage as a service therefore relies on the exhibition of cloud storage providers (CSP) competence in protecting stored data and transit data by providing security functionalities. It follows that sensitive data have to be encrypted prior to outsourcing for data privacy and combating unsolicited accesses. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in cloud computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. Thus, enabling an encrypted cloud data search service with secure index is of paramount importance. Considering the large number of data users and documents in cloud, it is crucial for the search service to allow multi-keyword query and provide effective data retrieval need.

## 2 Related Work

Most of the existing searchable encryption schemes use inverted index to speed up retrieval [2–4]. Inverted index though efficient is not without limitations; first limitation is that the inverted index data structure is not explicitly dynamic. The index has to be rebuilt in order to update (addition/deletion) keywords and documents. Even if the updates are handled, [4] a separate data structure like a delete array is used and the construction is complex allowing only limited updates. Secondly, updates leak information since the update information requires rebuilding inverted index at the CSP owned resources.

Moreover, for ranking, the pre-computed numerical document scores for a search keyword can be embedded along with the inverted index as postings [5]. However, it does not preserve the confidentiality of the indexed documents due to the posting elements added along with the index. It reveals the number of highly ranked documents for a given search keyword. Thus, there is a trade-off between confidentiality and retrieval effectiveness.

The above-mentioned problems are the motivation to design a top-k ranked searchable encryption for cloud data storage (TSED) which carries out the operations at the cloud storage itself without leaking information to the CSP. The proposed TSED is designed with the following objectives:

- An efficient dynamic inverted index with secure and efficient updates.
- Semantically secure index using probabilistic homomorphic encryption.
- Efficient top-k retrieval of documents.

The remainder of this paper is structured as follows: Section 3 introduces the system model of TSED. The construction of TSED is presented in Sect. 4 and performance analysis in Sect. 5. Section 6 presents the conclusion of TSED.

### 3 System Model

The architecture of TSED is illustrated in Fig. 1 which involves four entities namely the data owner or document owner (DO), cloud server or the CSP, secure coprocessor (SCP) and the data user (DU). The secure coprocessor [6] (like the IBM PCIe or the Freescale C29x) is assumed to reside at the cloud service providers' isolated execution environment. It is assumed that the cloud server and the secure coprocessor do not collude. The document set is encrypted using a private symmetric key. To enable searching through the document set, an encrypted searchable index per keyword (inverted index) is constructed from the document set. Both the encrypted document set and the encrypted index are outsourced to the cloud server. The data owner computes the term frequency and inverse document frequency and sends the encrypted score index per document (forward index) to the secure coprocessor. An authorized data user acquires a trapdoor corresponding to the search query from the data owner and submits the trapdoor or the search request to the CSP.

The CSP searches over the encrypted cloud storage and retrieves all the documents that match the search criteria. To reduce the communication cost, the CSP sends the trapdoor and an optional k obtained from the data user to the secure coprocessor to compute the top-k documents. The SCP computes the scores for the query keyword using the encrypted score index, ranks according to its relevance to the query and returns the top-k document identifiers to the CSP.

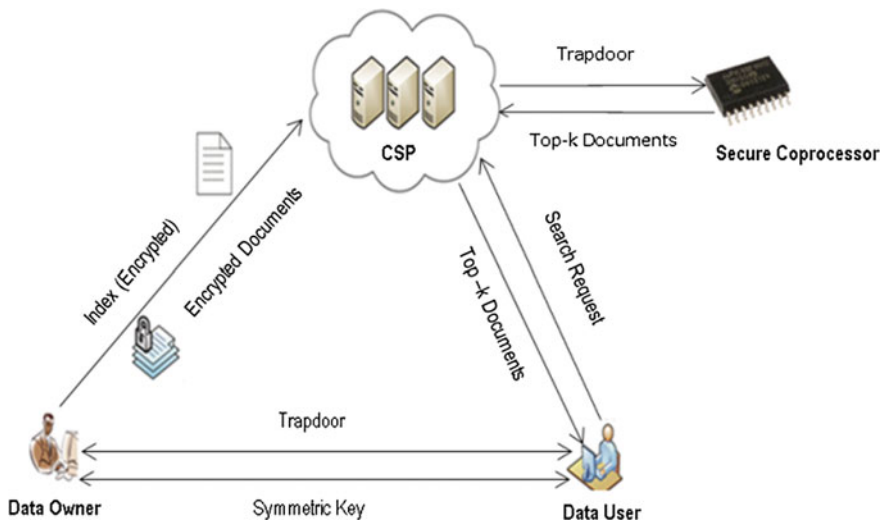


Fig. 1 TSED architecture



## 4 Construction of TSED

The construction of the TSED scheme aims to perform computations with the encrypted data or to modify the encrypted data using special functions in such a way that they are computed while preserving the privacy. Our construction relies on the two probabilistic encryption schemes namely Goldwasser-Micali [7] and Paillier cryptosystem [8]. The construction of the TSED scheme consists of the following algorithms:

**Setup:** Let  $\mathcal{D} = \{d_1, \dots, d_n\}$  where  $\mathcal{D}$  denotes the set of documents and  $d_i$  denotes the  $i$ th document. Let  $\mathcal{C} = \{C_1, \dots, C_n\}$ , where  $C_i = \text{Enc}(sk_{STD}, d_i)$  is the set of encrypted documents. Any standard symmetric key encryption algorithm proven to be a pseudo-random function like AES is assumed for  $\text{Enc}(sk_{STD}, d_i)$  and  $sk_{STD}$  is the secret key of the symmetric encryption algorithm. Let  $\mathcal{F} = \{f_1, \dots, f_n\}$  be the file identifiers of the encrypted documents given by  $f_i = \text{id}(C_i)$ . Let  $\mathcal{W} = \{w_1, \dots, w_{|\mathcal{W}|}\}_{\neq}$  be the set of distinct keywords of the document collection  $\mathcal{D}$ .

**KeyGen:** The TSED uses Goldwasser-Micali (GM) and Paillier (PL) homomorphic cryptosystems. Therefore, the KeyGen algorithm involves generation of two pairs of keys for each cryptosystems.

KeyGen for GM: Let  $n = pq$ , where  $p$  and  $q$  are distinct odd primes, and let  $m \notin \mathcal{QR}_n$  with  $\binom{m}{n} = -1$ . Let  $\mathcal{P} = \{0, 1\}$ ,  $\mathcal{C} = \mathcal{R} = \mathbb{Z}_n^*$ , and  $\mathcal{K} = \{(n, p, q, m)\}$  where  $n, p, q, m$  are described above. The key generation algorithm creates two distinct primes  $p$  and  $q$ , the value  $n = pq$ , as well as  $m \notin \mathcal{QR}_n$ . The tuple  $(n, m)$  is the public key of GM cryptosystem, denoted by  $pk_{GM}$ , and the value  $p$  is the private key of the GM cryptosystem, denoted as  $sk_{GM}$ . The keys are suffixed with their respective cryptosystem to differentiate them.

KeyGen for PL: The public key of Paillier cryptosystem is denoted as  $pk_{PL}$ , and the private key of Paillier cryptosystem is denoted as  $sk_{PL}$ . The mutually orthogonal vector is denoted by  $\mathcal{V}$ . The encryption and decryption operation of Paillier cryptosystem is denoted as  $\text{Enc}_{PL}(pk_{PL}, m)$  and  $\text{Dec}(sk_{PL}, c)$ , respectively.

**Enc<sub>GM</sub>(pk, m):** Given a public key  $pk_{GM} = (n, m)$  and a message  $x \in \mathcal{P}$ ,  $\text{Enc}(pk_{GM}, m)$  chooses a random value  $r \in \mathcal{R}$  and returns the ciphertext

$$c = m^x r^2 \text{ mod } n \quad (1)$$

**Dec<sub>GM</sub>(sk, c):** Given a private key  $sk_{GM} = p$  and a ciphertext,  $c \in \mathcal{C}$ ,  $\text{Dec}_{GM}(sk_{GM}, c)$  returns the following message

$$x = \begin{cases} 0 & \text{if } y \in \mathcal{Q} \mathcal{R}_n \\ 1 & \text{if } y \notin \mathcal{Q} \mathcal{R}_n \end{cases} \quad (2)$$

**BuildIndex**( $\mathcal{D}, \mathcal{W}, \mathbf{pk}_{GM}, \mathbf{pk}_{PL}, \mathcal{V}$ ): *BuildIndex* is executed by the document owner which takes input the data item  $w \in \mathcal{W}$ , the public key  $pk$  and outputs an index  $I_D$  which encodes a set of keywords,  $\mathcal{W}$ . Let  $|\mathcal{W}|$  be the cardinality of the set  $\mathcal{W}$ .

$$I_D = \sum_{i=1}^{|\mathcal{W}|} (v_i \cdot M_i \cdot S_i) + v_r \cdot r' \quad (3)$$

The data owner generates a binary data vector  $windex_i$  for every keyword  $w_i$ , where each binary bit  $windex_i[j] \in [0, 1] | j=1, \dots, |\mathcal{D}|, i=1, \dots, |\mathcal{W}|$  that represents the existence of the keyword  $w_i$  in the document  $d_j$  and computes the index denoted by  $I_D$ . The binary data vector is stored in a hash map (key-value pair) denoted as  $H_s(w_i)$ . Given a keyword  $w_i$ ,  $H_s(w_i)$  will give the  $windex_i$  for that respective keyword  $w_i$ . A secret vector  $v_i \in \mathcal{V}$  is assigned to every distinct keyword  $w_i \in \mathcal{W}$ , and  $r, r' \in \mathcal{R}$  is a number chosen at random. Let  $M_i = Enc_{PL}(pk_{PL}, w_i, r_{w_i})$  is the encryption of the keyword  $w_i$  and  $r_{w_i} \in \mathbb{Z}_n^* | \gcd(r_{w_i}, n) = 1$ . Let  $S_i = Enc_{GM}(pk_{GM}, H_s(w_i))$  be the encryption of binary data vector  $windex_i$ .

The document score is computed based on the term frequency ( $tf$ ) and inverse term frequency ( $idf$ ). The product of  $tf$  and  $idf$  is denoted by  $tfidf$ . The encrypted per document score index is denoted by  $Score_{d_j}$

$$Score_{d_j} = \sum_{i=1}^{|w_i \in d_j|} tfidf(w_i, d_j) \quad (4)$$

Each document's ( $d_j$ ) score for keywords  $w_i \in d_j$  is calculated as above and stored in the hash map, namely the score index,  $H_{ss}$  as given by  $H_{ss}(d_j)$ . The constructed score index is

$$H_{ss}(d_j) = Score_{d_j} = \sum_{i=1}^{|w_i \in d_j|} (v_i \cdot M_i \cdot tfidf(w_i, d_j)) + v_r \cdot r' \quad (5)$$

**TrapDoor**( $w_q, \mathbf{pk}_{PL}, \mathcal{V}$ ): The trapdoor generation algorithm takes a keyword  $w_i \in \mathcal{W}$ , the public key  $pk_{PL}$  as input and generates the trapdoor  $T_{w_q}$  to be used for searching a keyword in the encrypted index ( $I_D$ ). The trapdoor is constructed for each keyword in the query

$$T_{w_q} = v_{w_q} \cdot M_{w_q}^{-1} + v_r \cdot r'' \quad (6)$$

Let  $T_{w_q}$  be the trapdoor used to search for a keyword  $w_q$  in the encrypted index, where  $M_{w_q}^{-1} = Enc_{PL}(pk_{PL}, -w_q, r_q)$ ,  $-w_q$  is the negation of the keyword  $w_q$ ,

$r_q \in \mathbb{Z}_n^*$  such that  $r_q \cdot r_{w_q} \equiv 1 \pmod{n^2}$ ,  $v_{w_q} \in \mathcal{V}$  is the respective keyword's secret row vector,  $v_r \in \mathcal{V}$  is uniformly chosen random vector, and  $r'$  is a random nonce.

**Search**( $I_D, T_{w_q}, k$ ): This algorithm takes an index  $I_D$ , a trapdoor  $T_{w_q}$ , and  $k$  the number of documents to retrieve as input and outputs the set of top- $k$  file identifiers denoted by  $\mathcal{F}_{qk} = \{f_i\}_{\forall f_i \in w_q}$ .

$$\mathcal{E}_{\mathcal{F}_q} = \left( \sum_{i=1}^{|\mathcal{W}|} (v_i \cdot M_i \cdot S_i) + v_r \cdot r' \right) \cdot \left( v_{w_q} \cdot M_{w_q}^{-1} + v_r \cdot r' \right) = S_i \quad (7)$$

The same equation when  $w_q \notin \mathcal{W}$ , the search algorithm returns  $\perp$ . Therefore,  $\mathcal{E}_{\mathcal{F}_q}$  results in either  $\perp_{w_q \notin \mathcal{W}}$  or  $S_i = \text{Enc}_{GM}(pk_{GM}, \text{index}_i)$ , the encryption of the  $\text{index}_i$  or  $\text{index}_q$  (the encryption of the binary data vector). The search algorithm then invokes the SCP to execute *topk* algorithm in order to retrieve *topk* documents that match the specified search query, for  $k$  being specified by the user.

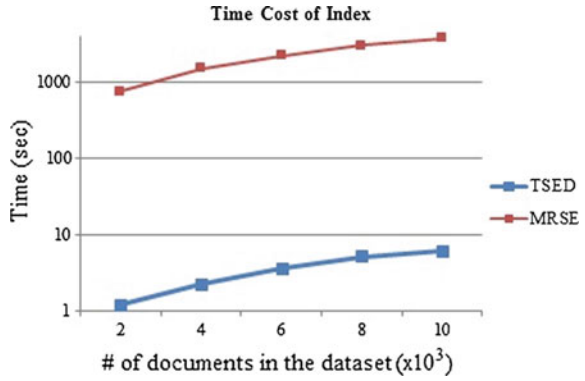
**Topk**( $k, sk_{GM}, \mathcal{E}_{\mathcal{F}_q}, T_{w_q}, H_{ss}$ ): This algorithm takes a value  $k$ , encryption of  $\text{index}_q$  computed by the search algorithm and a trapdoor  $T_{w_q}$ , generated by the DO. The *topk* algorithm executes at the SCP and decrypts  $\mathcal{E}_{\mathcal{F}_q}$  to get binary data vector  $\text{index}_q$  for the keyword  $w_q$ .  $\text{index}_q$  is the binary data vector in which the binary bits are set if that document contains the keyword  $w_q$ . The *topk* algorithm finds the score for the each document for which  $\text{index}_q$  bit is set by the operation  $(H_{ss}(d_x) \cdot T_{w_q})$  and then sorts the obtained scores. Then, it returns the top  $k$  documents from the sorted result to the CSP.

**Index Update**( $I_D, Y_{w_u}$ ): This algorithm takes an index  $I_D$ , a trapdoor for update  $Y_{w_u}$ , and the public key  $pk_{PL}$  as input, update document or keyword and outputs an updated index. The process of adding a keyword uses simple addition and deletion of vectors can be represented as follows:

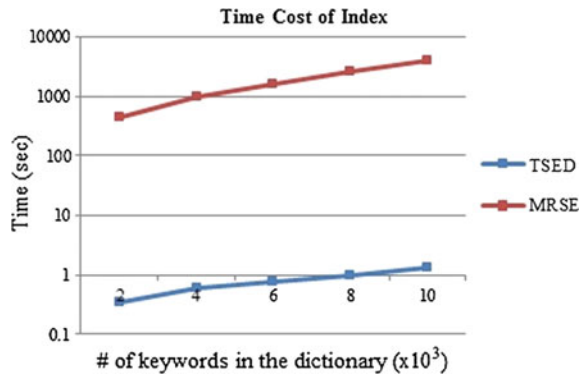
$$I_D = I_D - Y_{w_u} + Y'_{w_u} \quad (8)$$

where the old subindex is denoted by  $Y_{w_u} = v_u \cdot M_u \cdot S_u$ , and the new subindex that needs to updated with index is denoted by  $Y'_{w_u} = v_u \cdot M_u \cdot S'_u$ .

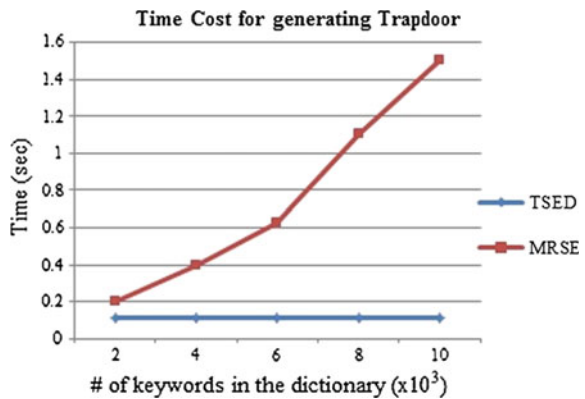
**Fig. 2** Time cost of index (varying dataset)



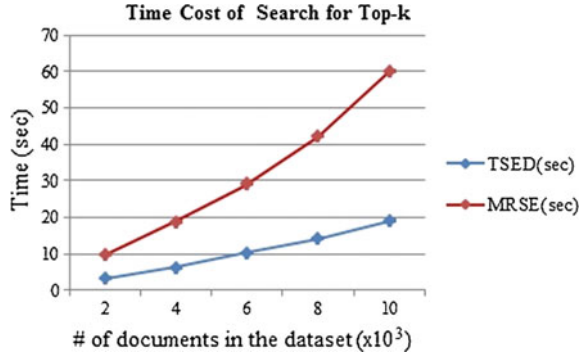
**Fig. 3** Time cost of index (varying dictionary)



**Fig. 4** Time cost of trapdoor (varying dataset)



**Fig. 5** Time cost of search (varying dataset)



## 5 Performance Analysis

The performance of TSED is analyzed with real-world dataset and compared with multi-keyword ranked searchable encryption (MRSE) [9]. TSED is implemented using Java language on a Windows server with Intel Xeon Processor 2.30 GHz. The performance of TSED is evaluated for the efficiency of index construction, trapdoor building, and search operation.

The time cost of building the index is found by varying the documents and fixing the size of the dictionary  $|\mathcal{W}| = 4000$  as seen in Fig. 2 and varying the dictionary and keeping the document constant  $|D| = 1000$  as in Fig. 3. Figure 4 shows the time cost for trapdoor generation. The time complexity of search operation in TSED is constant time, i.e.,  $O(1)$  without top-k and  $O(|\mathcal{F}_{qk}|)$  with top-k. The complexity of MRSE is  $(|D||W|)$ , as similarity scores for all the documents are computed as seen in Fig. 5. Experimental analysis shows that TSED outperforms MRSE with improved search.

## 6 Conclusion

A practical multi-keyword top-k ranked search over encrypted data based on searchable dynamic inverted index scheme is proposed using the probabilistic encryption scheme, namely Goldwasser-Micali (GM) and Paillier encryption. The architecture of TSED is designed in such a way that the top-k ranking is computed by the secure coprocessor. This architecture ensures that neither the CSP nor the SCP learns anything about the data involved other than the allowed information leakage as described in the security model. Thus, TSED cannot be compromised by timing-based side channel attacks that try to differentiate queries based on query time. Experimental analysis shows that TSED outperforms MRSE and allows dynamic update of index with improved search.

## References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R.H.: Above the Clouds: A Berkeley View of Cloud Computing. University of California, Berkeley, Technical report UCB. 07-013 (2009)
2. Naveed, M., Prabhakaran, M., Gunter, C.A.: Dynamic Searchable Encryption via Blind Storage. In: 2014 IEEE Symposium on Security and Privacy, pp. 639–654 (2014)
3. Cash, D., Jaeger, J., Jarecki, S., Jutla, C., Krawczyk, H., Roşu, M.-C., Steiner, M.: Dynamic searchable encryption in very-large databases: data structures and implementation. In: Proceedings of the 2014 Network and Distributed System Security Symposium, pp. 1–32 (2014)
4. Kamara, S., Papamanthou, C., Roeder, T.: Dynamic Searchable Symmetric Encryption, pp. 1–24
5. Zerr, S., Olmedilla, D., Nejdil, W., Siberski, W.: Zerber +R. top-k retrieval from a confidential index. In: Proceedings 12th International Conference on Extending Database Technology. Advances in Database Technology—EDBT '09, p. 439 (2009)
6. Baldimtsi, F., Ohrimenko, O.: Sorting and Searching Behind the Curtain, pp. 1–25
7. Goldwasser, S., Bellare, M.: Lecture notes on cryptography. Cryptogr. Comput. Secur. 1–289 (2008)
8. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. Int. Conf. Theory. (1999)
9. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**, 222–233 (2014)

# An Efficient Forward Secure Authenticated Encryption Scheme with Ciphertext Authentication Based on Two Hard Problems



Renu Mary Daniel, Elijah Blessing Rajsingh and Salaja Silas

**Abstract** Authenticated encryption is a cryptographic technique that concurrently establishes message confidentiality, integrity, authenticity and non-repudiation. In this paper, an efficient authenticated encryption scheme is proposed, based on the hardness of the integer factorization problem and the discrete logarithm problem on conic curves over a ring  $Z_n$ . The protocol provides forward secrecy in case the sender's private keys are compromised and supports public verifiability, as well as, ciphertext authentication by an external verifier, without full decryption. Hence, the protocol can be used for secure data sharing in untrusted cloud environments. Several attack scenarios against the scheme are analysed to confirm its validity as an authenticated encryption protocol. The security criteria are satisfied, as long as either one of the hardness assumptions hold. The scheme is implemented over conic curves, which possess interesting characteristics like effective message encoding and decoding, easily computable point operations and inverses.

**Keywords** Authenticated encryption · Conic curve · Ciphertext authentication · Forward secrecy · Public verifiability

## 1 Introduction

According to the conventional cryptographic approach, message confidentiality, authenticity and integrity are guaranteed by first signing the message using the sender's private key and then encrypting the message–signature pair, using a one-time session key. Further, the session key is encrypted with the intended recipient's public key. The encrypted message and random key are then sent to the receiver, who retrieves the session key using the corresponding private key for full decryption and sender verification. In 1997, Zheng [1] proposed an authenticated encryption scheme called signcryption, which cleverly combines the functionalities of both encryption

---

R. M. Daniel (✉) · E. B. Rajsingh · S. Silas  
Karunya University, Coimbatore 641114, Tamil Nadu, India  
e-mail: renumarydaniel@karunya.edu.in

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_11](https://doi.org/10.1007/978-981-10-7200-0_11)

and digital signature in a single logical step, significantly increasing efficiency, when compared to the traditional “signature followed by encryption” approach. The protocol was based on the hardness of computing discrete logarithm problem (DLP) over a finite field. Later, Zheng and Imai proposed a variant of the scheme based on the elliptic curve analogue of discrete logarithm problem (ECDLP) (see [2, 3]). A hoard of other signcryption protocols followed, with properties like forward secrecy [4, 5], public verifiability [6] and ciphertext authentication [7]. Mohamed and Elkamchouchi proposed a forward secure signcryption scheme with ciphertext authentication and public verifiability [8]. But the security of all these protocols is based on individual hardness assumptions. Elkamchouchi et al. [9] proposed a proxy signcryption scheme based on multiple hardness assumptions in finite fields; however, they used a composite modulus comprising of four primes. This reduces efficiency, since the modulus size should be at least 4096 bits, to resist factoring attacks by elliptic curve method [10, 11]. The protocol lacks ciphertext authentication, hence the original message must be revealed to the external entity for verification, in case of a dispute. In this paper, a novel efficient signcryption scheme is proposed based on conic curve discrete logarithm problem (CCDLP) and integer factorization problem (IFP) that ensures forward secrecy, public verifiability and ciphertext authentication in addition to confidentiality, authenticity and non-repudiation. The probability that an adversary simultaneously solves two hard problems is negligible, hence the proposed scheme offers better security. The protocol is highly suitable for securing data transmissions in cloud environments [12].

**Outline of the paper:** The rest of the paper is organized as follows; Sect. 2 gives a brief account of conic curve cryptography. The proposed scheme is presented in Sect. 3. Section 4 provides the detailed security analysis. The performance of the proposed scheme is analysed in Sect. 5. Section 6 documents the concluding remarks.

## 2 Conic Curve Cryptography

Conic curve groups over a finite field have been used to design analogues of RSA cryptosystem [13, 14], Diffie–Hellman key exchange protocol [15], El-Gamal encryption scheme [16] and various signature schemes [17–21], owing to the simplicity of group operations, when compared to their elliptic curve counterparts. Dai et al. in 2001, proved that DLP over a conic curve group is no stronger than the DLP over the multiplicative group of a finite field [22]. Nevertheless, conic curve groups are still widely researched due to several interesting characteristics like effective message encoding and decoding, easily computable point operations and inverses.



## 2.1 Conic Curve Group over a Finite Field $F_p$

Let  $F_p$  be a finite field of order  $p$ , where  $p$  is an odd prime. Then, the multiplicative group  $F_p^*$  can be represented as  $F_p/\{0\}$ . A conic curve denoted as  $C(a, b)$  over the finite field  $F_p$  is the solution set in the affine plane  $A^2(F_p)$ , with equation

$$C(F_p) : y^2 = ax^2 - bx, (a, b) \in F_p^*. \quad (1)$$

Apparently, the origin  $O(0, 0)$  satisfies Eq. (1). For  $x \neq 0$ , let  $t = yx^{-1}$ , substituting  $y = xt$  in Eq. (1), we get,

$$b = x(a - t^2), (a, b) \in F_p^*. \quad (2)$$

If  $a \neq t^2$ , from Eq. (2),

$$\begin{aligned} b(a - t^2)^{-1} &= x \\ bt(a - t^2)^{-1} &= y \end{aligned} \quad (3)$$

Consider a set,  $H = \{t \in F_p : t^2 \neq a\} \cup \{O\}$ , then the mapping  $P : H \rightarrow C(F_p)$  is a bijection, where  $P(t) = (x_t, y_t)$ ,  $b(a - t^2)^{-1} = x_t$ ,  $bt(a - t^2)^{-1} = y_t$ ,  $t \neq O$  and  $O = (0, 0)$ .  $()^{-1}$  denotes the multiplicative inverse. Consider an addition operation  $\otimes$  on elements of the curve  $C(F_p)$  defined by Eqs. (4) and (5).

$$\forall P(t) \in C(F_p), P(O) \otimes P(t) = P(t) \otimes P(O) = P(t) \quad (4)$$

For any  $P(t_1), P(t_2) \in C(F_p)$ ,  $t \in H$  and  $t \neq O$ ,

$$P(t_1) \otimes P(t_2) = P(t_3) \quad (5)$$

$$t_3 = \begin{cases} (t_1 t_2 + a)(t_1 + t_2)^{-1}, & (t_1 + t_2 \neq 0) \\ O, & (t_1 + t_2 = 0) \end{cases} \quad (6)$$

$$-P(O) = P(O), -P(t) = P(-t)$$

From (4)–(6), it is evident that the operation  $\otimes$  is associative over  $C(F_p)$ . The conic curve  $(C(F_p), \otimes, P(O))$  forms a finite abelian group under the operation  $\otimes$ . Considering the Legendre symbol  $\left(\frac{a}{p}\right)$ , the cardinality of  $C(F_p)$  can be defined as  $|C(F_p)| = p - \left(\frac{a}{p}\right)$ . It can be proved that  $\forall P(t) \in C(F_p)$ ,  $|C(F_p)|P(t) = P(O)$  and  $kP(t) = \{P(t) \otimes \dots \otimes P(t)\}$ ,  $k$  times. Any message  $m \in H/\{O\}$  can be encoded on the curve  $C(a, b)$ , as  $P(m) = (x_m, y_m)$ , where  $b(a - m^2)^{-1} = x_m$  and  $bm(a - m^2)^{-1} = y_m$ . The message can be decoded as  $x_m^{-1}y_m \pmod{p} = m$ .

## 2.2 Conic Curve over a Ring $Z_n$

A conic curve over a ring  $Z_n$  can be represented by the equation  $C_n(a, b) : y^2 \equiv ax^2 - bx \pmod{n}$ ,  $(a, b) \in Z_n$ , where  $n = pq$ , such that  $n$  is relatively prime to both  $a$  and  $b$ . The numbers  $p$  and  $q$  are large odd primes, such that  $p + 1 = 2r$ ,  $q + 1 = 2s$  and  $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$ . Then, the order  $C_n(a, b)$  is obtained as  $N_n = \text{lcm}(|C(F_p)|, |C(F_q)|) = \text{lcm}(2r, 2s) = 2rs$ . The CCDLP is defined as the problem of computing  $k \in Z_{N_n}^*$ , given a point  $P_1 = (x_{p_1}, y_{p_1}) \in C_n(a, b)$  and another point  $P_2 = kP_1 \pmod{n}$ .

## 3 Proposed Scheme

The protocol is based on the conic curve congruence equation described in the previous section. All communicating entities in the system decide on a security parameter  $\lambda$ , which determines the group order and size of keys. For instance, in a cloud environment, these parameters are chosen by the Cloud Service Provider (CSP). Assuming that cloud user A is the data owner and user B is the intended recipient, the protocol is executed as follows:

### 3.1 Key Extraction Phase

1. User A chooses large secure odd primes  $p_a$  and  $q_a$ .
2. Computes  $n_a = (p_a \cdot q_a)$  and  $N_{n_a} = \text{lcm}(|C(F_{p_a})|, |C(F_{q_a})|)$ .
3. User A chooses  $x_a \in Z_{N_{n_a}}^*$  and computes  $Y_a = x_a G_a \pmod{n_a}$  over the base point  $G_a \in C_{n_a}(a, b)$ . Here  $x_a$  is the secret key, and  $Y_a$  is the corresponding public key.
4. Another integer  $e_a$  is chosen, such that  $e_a$  is coprime to  $N_{n_a}$ .
5. User A computes an integer  $d_a$ , such that  $e_a d_a \equiv 1 \pmod{N_{n_a}}$ .
6. User A's public parameters are  $(n_a, G_a, Y_a, e_a)$  and the secret key is the tuple  $(x_a, d_a, N_{n_a})$ .

User B follows the same procedure to generate the public parameters  $(n_b, G_b, Y_b, e_b)$  and secret key  $(x_b, d_b, N_{n_b})$ . It is assumed that the public keys of all entities are certified by a trusted certificate authority. In the proposed scheme,  $\text{hash}(\cdot)$  denotes a secure hash function like SHA-2,  $E_k(\cdot)$ ,  $D_k(\cdot)$  denotes symmetric encryption and decryption respectively using AES, under the key  $k$ . The notation  $KH_k(\cdot)$  denotes a keyed hash function, using key  $k$ .

### 3.2 Signcryption

User A (data owner) signcrypts the message  $m$  as follows:

1. User A chooses a random integer  $v$  such that,  $vd_a \in Z_{n_b}^*$ .
2. Compute  $hash(\alpha) = (k_1, k_2)$ , where  $\alpha \equiv vd_a G_b \pmod{n_b}$ , if  $\alpha = 0$ , return to step 1.
3. Compute  $c = E_{k_1}(P_b(M))$ , where  $P_b(M)$  is the mapping of  $m$  to a point on the curve  $C_{n_b}(a, b)$ .
4.  $r = KH_{k_2}(P_b(M))$
5.  $s = vd_a e_b Y_b \pmod{n_b}$
6.  $\tau = d_a x_a^{-1} r \pmod{N_{n_a}^*}$

User A stores the tuple  $(c, r, s, \tau)$  in the cloud repository. Upon successful authentication, CSP transmits the tuple  $(c, s, \tau)$  to the legitimate receiver B.

### 3.3 Unsigncryption

User B successfully unsigncrypts with the ciphertext tuple  $(c, s, \tau)$  as follows:

1. Compute  $d_b x_b^{-1} s = vd_a G_b \pmod{n_b} = \alpha$
2. Compute  $hash(\alpha) = (k_1, k_2)$ ,
3. Decrypt  $c$  as,  $D_{k_1}(c) = P_b(M)$
4. Compute  $KH_{k_2}(P_b(M)) = r$
5. Compute  $e_a \tau Y_a \pmod{n_a} = T_1$
6. Check if  $rG_a \pmod{n_a} = T_1$ , if the equality check fails, the ciphertext is discarded, else, the message  $m$  is recovered from  $P_b(M)$ , using the decoding algorithm.

However, it is observed that if the entire ciphertext tuple  $(c, r, s, \tau)$  is forwarded to user B, he can reject the ciphertext, if it is tampered or forged. This can be done using a simple two-step verification process, without decrypting the entire message.

1. Compute  $e_a \tau Y_a \pmod{n_a} = T_1$
2. Check if  $rG_a \pmod{n_a} = T_1$ , if the equality check fails, the ciphertext is discarded.

### 3.4 Correctness of the Protocol

During ciphertext validation, the computation  $e_a \tau Y_a \pmod{n_a} = e_a (d_a r x_a^{-1}) x_a G_a \pmod{n_a} = (e_a d_a) r (x_a^{-1} x_a) G_a \pmod{n_a} = rG_a \pmod{n_a} = T_1$ . This computation authenticates the ciphertext sender. Both the public keys of the sender will be

applied to retrieve  $T_1$  for further verification. During unisgncription, the intended receiver B obtains the one-time session key, by applying both the private keys on the token  $s$ . Only user B can retrieve  $\alpha$ , using the private key components  $(d_b, x_b^{-1})$ . User B performs the computation  $d_b x_b^{-1} s = d_b x_b^{-1} v d_a e_b Y_b \pmod{n_b} = (v d_a) d_b x_b^{-1} e_b x_b G_b \pmod{n_b} = (v d_a) d_b e_b x_b^{-1} x_b G_b \pmod{n_b} = (v d_a) G_b \pmod{n_b} = \alpha$ .

## 4 Security Analysis

This section illustrates how the proposed system withstands four common attack scenarios by an adversary. Security features of the proposed scheme, namely public verifiability, ciphertext authentication, forward secrecy, non-repudiation, message integrity and confidentiality are also substantiated.

### 4.1 Direct Attack Against a Legitimate User

An adversary might try to gain the private keys of a legitimate user, from the corresponding public keys. But this requires solving simultaneously CCDLP and IFP. Any message intended to a legitimate user can only be decrypted using both the private keys of the receiver. A masquerader cannot create valid ciphertexts, because the ciphertext component cannot be computed without both the private keys of the sender. To prevent integer factorization attacks, the modulus size must be at least 1024 bits. The component prime numbers  $p$  and  $q$  must be randomly chosen and should be roughly of the same size, for example 512 bits.

**Integer Factorization Attack:** Assume that an adversary successfully solves IFP and obtains the private key component  $d_b$ , corresponding to public key  $e_b$  of user B. The adversary applies the known private key component  $d_b$  on ciphertext the component  $s$ . The computation  $d_b s = d_b v d_a e_b Y_b \pmod{n_b} = (v d_a) d_b e_b x_b G_b \pmod{n_b} = (v d_a) x_b G_b \pmod{n_b} \neq (v d_a) G_b \pmod{n_b}$ . Hence, he cannot retrieve the session key  $\alpha$ . Similarly, while forging a ciphertext, the component  $\tau$  will not be well-formed as the adversary has no clue about  $x_b^{-1}$ .

**Conic Curve Discrete Logarithm Attack:** Assume that the adversary successfully solves the CCDLP and obtains the secret key  $x_b$ , corresponding to the public key component  $Y_b = x_b G_b \pmod{n_b}$ . The adversary applies the known private key component  $x_b$  on ciphertext the component  $s$ . The computation  $x_b^{-1} s = x_b^{-1} v d_a e_b Y_b \pmod{n_b} = v d_a e_b x_b^{-1} x_b G_b \pmod{n_b} = v d_a e_b G_b \pmod{n_b} \neq \alpha$ . The blinding factor  $e_b$  cannot be eliminated without solving IFP, hence decryption fails. Also, ciphertexts cannot be forged, as the component  $\tau$  cannot be computed without the knowledge of  $d_b$ .

## 4.2 Impersonation Attack

An analogue of the following attack [23] was identified against Yang et al. [24] authenticated encryption scheme by Chaudhry et al. During the attack, another cloud user C impersonates user A and performs a message signcryption for recipient B. User C (impersonating user A) signcrypts the message  $m$  as follows:

1. User C chooses a random integer  $v$  such that,  $v \in Z_{n_b}^*$ .
2. Compute  $hash(\alpha) = (k_1, k_2)$ , where  $\alpha \equiv vG_b \pmod{n_b}$ , if  $\alpha = 0$ , return to step 1.
3.  $c = E_{k_1}(P_b(M))$ , where  $P_b(M)$  is the mapping of  $m$  to a point on the curve  $C_{n_b}(a, b)$ .
4.  $r = KH_{k_2}(P_b(M))$
5.  $s = ve_bY_b \pmod{n_b}$

The sender C forwards the tuple  $(c, r, s)$  to the cloud. Suppose, the system requires only, these ciphertext components. User B successfully unsigncrypts with the ciphertext tuple  $(c, r, s)$  as follows:

1. Compute  $d_bx_b^{-1}s = vG_b \pmod{n_b} = \alpha$
2. Compute  $hash(\alpha) = (k_1, k_2)$ ,
3. Recover  $P_b(M) = D_{k_1}(c)$
4. Check if  $KH_{k_2}(P_b(M)) = r$

Yang et al. system fails since the receiver is not using the sender's public key anywhere during unsigncryption. However in the proposed scheme, the sender must successfully compute another ciphertext component  $\tau = d_ax_a^{-1}r \pmod{N_a^*}$ , which cannot be created without the knowledge of both the private keys  $(d_a, x_a)$  of the sender A. Also, any previously stored  $\tau$  cannot be used for the attack, and since on decryption, it will not match with the ciphertext component  $r$  of the forged ciphertext.

## 4.3 Public Verifiability and Ciphertext Authentication

Any external entity (like the CSP) can verify the ciphertext authenticity and integrity, using the alleged sender's public key components. The receiver need not reveal the original message for sender verification. In case of a dispute over a ciphertext tuple  $(c, s, \tau)$ , the CSP retrieves the complete tuple  $(c, r, s, \tau)$ . The CSP verifies the ciphertext using the elements  $(r, \tau)$ , as follows:

1. Compute  $e_a\tau Y_a \pmod{n_a} = T_1$ .
2. Check if  $rG_a \pmod{n_a} = T_1$ , if the equality check succeeds, the message was indeed sent by user A, the owner of the public key pair  $(e_a, Y_a)$ .

If the external verifier is a firewall, the entire message tuple  $(c, r, s, \tau)$  is sent for initial verification. If the verification succeeds, it forwards the tuple  $(c, s, r)$  to user B. Receiver B unsigncrypts with fewer steps as follows:

1. Compute  $d_b x_b^{-1} s = v G_b \pmod{n_b} = \alpha$
2. Compute  $hash(\alpha) = (k_1, k_2)$ ,
3. Recover  $P_b(M) = D_{k_1}(c)$
4. Check if  $KH_{k_2}(P_b(M)) = r$

#### 4.4 Forward Secrecy

Assume that the sender A's private key components are leaked. Hence, an adversary is in possession of the components  $(x_a, d_a, N_{n_a})$ . Still, the adversary will not be able to decrypt any of the messages previously signcrypted by user A, from the stored tuples of the form  $(c, r, s, \tau)$ . The adversary cannot compute the randomly chosen value  $v$ , hidden in  $s$ , without knowing both the private keys of the recipient.

#### 4.5 Unforgeability and Non-repudiation

It is evident that no adversary can forge a message without obtaining all the private key components of the sender. But this requires simultaneously solving CCDLP and IFP. No probabilistic polynomial time adversary can solve these hardness assumptions concurrently with non-negligible probability. Hence, it is also impossible for the original sender of the message to repudiate the ciphertext created by him.

#### 4.6 Integrity and Confidentiality

The integrity of each ciphertext can be validated by just using the ciphertext components  $(r, \tau)$ . The integrity of the message is validated by the receiver using steps 4–6 in Sect. 3.3. Also, only the receiver can decrypt the message using both the private key components, ensuring confidentiality.

### 5 Efficiency of the Scheme

The proposed scheme is efficient when compared to the signcryption schemes designed based on DLP [1] and ECDLP [6, 8]. The computations in the new signcryption scheme do not involve group exponentiations or comparatively expensive elliptic curve point operations. The computation cost of inexpensive operations such as hashing, symmetric encryption and decryption is ignored. The time complexity of the protocol is mainly attributed by simple conic curve scalar multiplication (repeated

**Table 1** Comparison of the computational complexity and hardness assumptions of the proposed scheme with Mohamed and Elkamchouchi protocol [8], as well as Elkamchouchi et al. scheme [9].  $T_{CC-sm}$ —Conic curve scalar multiplication,  $T_{mul}$ —Modular multiplication,  $T_{Exp}$ —Group exponentiation,  $T_{EC-sm}$ —Elliptic curve scalar multiplication

Criteria	Our scheme	Reference [9]	Reference [8]
Signcryption	$2T_{CC-sm} + 1T_{mul}$	$2T_{Exp} + 1T_{mul}$	$2T_{EC-sm} + 1T_{mul}$
Complete unsigncryption	$3T_{CC-sm}$	$3T_{Exp}$	$3T_{EC-sm}$
Ciphertext verification only	$2T_{CC-sm}$	Not supported	$2T_{EC-sm}$
Hardness assumptions	CCDLP and IFP	DLP and IFP	ECDLP only

application of conic curve operation  $\otimes$ ) denoted by  $T_{(CC-sm)}$  and modular multiplication  $T_{mul}$ . The communication cost is nearly  $2|n|$ . Table 1 provides an overview of the computation cost of the system.

## 6 Conclusion

This paper proposes a novel authenticated encryption scheme based on two hardness assumptions, namely conic curve discrete logarithm problem and integer factorization problem. In addition to properties like confidentiality, authenticity, non-repudiation and integrity, the protocol ensures forward secrecy, ciphertext authentication and public verifiability. An external verifier like firewall can reject the ciphertext in case it is tampered or forged, without decrypting the message or forwarding it to the receiver, saving computation cycles. The original message need not be revealed to a verifier in case of a dispute. Even if a legitimate user’s private keys are compromised, all the previously encrypted messages remain confidential. The protocol is designed on conic curve groups over finite fields, hence the main computation is simple scalar multiplication on conics. Messages can be easily coded and decoded on conic curves. The protocol can be used in securing cloud environments, e-commerce transactions and network communications.

**Acknowledgements** This work was funded by Visvesvaraya PhD Scheme for Electronics and IT, Ministry of Electronics and Information Technology, Government of India.

## References

1. Zheng, Y.: Digital signcryption or how to achieve cost (signature & encryption) cost (signature)+ cost (encryption). In: Annual International Cryptology Conference, pp. 165–179. Springer, Berlin Heidelberg (1997). <https://doi.org/10.1007/BFb0052234>

2. Zheng, Y., Imai, H.: How to construct efficient signcryption schemes on elliptic curves. *Informat. Process. Lett.* **68**, 227–253. Elsevier (1998). [https://doi.org/10.1016/S0020-0190\(98\)00167-7](https://doi.org/10.1016/S0020-0190(98)00167-7)
3. Zheng, Y., Imai, H.: Efficient signcryption schemes on elliptic curves. Citeseer (1996). 10.1.1.130.4261
4. Hwang, R.J., Lai, C.H., Su, F.F.: An efficient signcryption scheme with forward secrecy based on elliptic curve. *Appl. Mathemat. Comput.* **167**, 870–881. Elsevier (2005). <https://doi.org/10.1007/s11042-014-2283-9>
5. Toorani, M., Beheshti, A.A.: An elliptic curve-based signcryption scheme with forward secrecy. arXiv preprint [arXiv:1005.1856](https://arxiv.org/abs/1005.1856) (2010). <https://doi.org/10.3923/jas.2009.1025.1035>
6. Xiang-xue, L., Ke-fei, C., Shi-qun, L.: Cryptanalysis and improvement of signcryption schemes on elliptic curves. *Wuhan Univ. J. Nat. Sci.* **10**(1), 231–234 (2005). <https://doi.org/10.1007/BF02828657>
7. Chow, S.S., Yiu, S.M., Hui, L.C., Chow, K.P.: Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity. In: *International Conference on Information Security and Cryptology*, pp. 352–369. Springer Berlin Heidelberg (2003). [https://doi.org/10.1007/978-3-540-24691-6\\_26](https://doi.org/10.1007/978-3-540-24691-6_26)
8. Mohamed, E., Elkamchouchi, H.: Elliptic curve signcryption with encrypted message authentication and forward secrecy. *Int. J. Comput. Sci. Netw. Secur.* **9**(1), 395–398 (2009)
9. Elkamchouchi, H., Nasr, M., Ismail, R.: A new efficient strong proxy signcryption scheme based on a combination of hard problems. In: *IEEE International Conference on Systems, Man and Cybernetics, SMC 2009*. IEEE, pp. 5123–5127 (2009). <https://doi.org/10.1109/ICSMC.2009.5346018>
10. Hinek, M.J.: On the security of multi-prime RSA. *J. Math. Cryptology* **2**(2), 117–147 (2008). 0.1515/JMC.2008.006
11. Ciet, M., Koeune, F., Laguillaumie, F., Quisquater, J.J.: Short private exponent attacks on fast variants of RSA. UCL Crypto Group Technical Report Series CG-2002/4, University Catholique de Louvain (2002). doi:10.1.1.12.9925
12. Sun, Y., Zhang, J., Xiong, Y., Zhu, G.: Data security and privacy in cloud computing. *Int. J. Distrib. Sens. Netw.* (2014)
13. Chen, Z.G., Song, X.X.: A public-key cryptosystem scheme on conic curves over  $Z_n$ . In: *2007 International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2183–2187. IEEE (2007). <https://doi.org/10.1109/ICMLC.2007.4370507>
14. Bellini, E., Murru, N.: An efficient and secure RSA-like cryptosystem exploiting Rdei rational functions over conics. *Finite Fields Appl.* **39**, 179–194 (2016). <https://doi.org/10.1016/j.ffa.2016.01.011>
15. Zheng Fu, C.: A public key cryptosystem based on conic curves over finite field  $F_p$ . *ChinaCrypt*, pp. 45–49, Science Press (1998)
16. Zhang, D., Liu, M., Yang, Z.: Zero-knowledge proofs of identity based on ELGAMAL on conic. In: *IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, pp. 216–223. IEEE (2004). <https://doi.org/10.1109/CEC-EAST.2004.77>
17. Tahat, N.M.: A new conic curve digital signature scheme with message recovery and without one-way hash functions. *Ann. Univ. Craiova-Math. Comput. Sci. Ser.* **40**(2), 148–153 (2013)
18. Shi, Y., Xiong, G.Y.: An undetachable threshold digital signature scheme based on conic curves. *Appl. Math. Inf. Sci.* **7**(2), 823–828 (2013). <https://doi.org/10.12785/amis/070254>
19. Song, X., Chen, Z.: An efficient conic curve threshold digital signature. In: *Proceedings of the 3rd WSEAS International Conference on Circuits, Systems, Signal and Telecommunications*, pp. 149–153 (2009)
20. Dong, X., Qian, H., Cao, Z.: Provably secure RSA type signature based on conic curve. *Wirel. Commun. Mob. Comput.* **9**(2), 217–225 (2009). <https://doi.org/10.1002/wcm.602>
21. Lu, R.X., Cao, Z.F., Zhou, Y.: Threshold undeniable signature scheme based on conic. *Appl. Math. Comput.* **162**(1), 165–177 (2005). <https://doi.org/10.1016/j.amc.2003.12.084>
22. Dai, Z.D., Ye, D.F., Pei, D.Y., Yang, J.H.: Cryptanalysis of ElGamal type encryption schemes based on conic curves. *Electron. Lett.* **37**(7), 426 (2001). <https://doi.org/10.1049/el:20010272>



23. Chaudhry, S.A., Farash, M.S., Naqvi, H., Sher, M.: A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. *Electron. Commer. Res.* **16**(1), 113–139 (2016). <https://doi.org/10.1007/s10660-015-9192-5>
24. Yang, J.H., Chang, Y.F., Chen, Y.H.: An efficient authenticated encryption scheme based on ECC and its application for electronic payment. *Inf. Technol. Control* **42**(4), 315–324 (2013). <https://doi.org/10.5755/j01.itc.42.4.2150>

# Clustered Queuing Model for Task Scheduling in Cloud Environment



Sridevi S. and Rhymend Uthariaraj V.

**Abstract** With the advent of big data and Internet of Things (IoT), optimal task scheduling problem for heterogeneous multi-core virtual machines (VMs) in cloud environment has garnered greater attention from researchers around the globe. The queuing model used for optimal task scheduling is to be tuned according to the interactions of the task with the cloud resources and the availability of cloud processing entities. The queue disciplines such as First-In First-Out, Last-In First-Out, Selection In Random Order, Priority Queuing, Shortest Job First, Shortest Remaining Processing Time are all well-known queuing disciplines applied to handle this problem. We propose a novel queue discipline which is based on k-means clustering called clustered queue discipline (CQD) to tackle the above-mentioned problem. Results show that CQD performs better than FIFO and priority queue models under high demand for resource. The study shows that: in all cases, approximations to the CQD policies perform better than other disciplines; randomized policies perform fairly close to the proposed one and, the performance gain of the proposed policy over the other simulated policies, increase as the mean task resource requirement increases and as the number of VMs in the system increases. It is also observed that the time complexity of clustering and scheduling policies is not optimal and hence needs to be improved.

**Keywords** Cloud computing • Load balancing • Clustering Queuing discipline

---

Sridevi S. (✉) · Rhymend Uthariaraj V.  
Anna University, Chennai, India  
e-mail: sridevioct90@gmail.com

Rhymend Uthariaraj V.  
e-mail: rhymend@annauniv.edu

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_12](https://doi.org/10.1007/978-981-10-7200-0_12)

## 1 Introduction

The task scheduling problem in cloud environment is a well-known NP hard problem where the queuing strategy adopted to schedule tasks plays a vital role. The existing queuing disciplines follow strategies that are not well suited for cloud environments. Based on the extensive study carried out, it is found that the queuing discipline that is best suitable for cloud environment should possess the following characterization: must possess multiple queues that can be quickly forwarded to appropriate nodes [1]; must have single entry point for tasks; multiple points of VM buffer and after processing finally exiting the system; joint probability distributions are to be computable; fair queuing with minimum waiting time of tasks and maximum utilization of virtual machines (VMs).

The primary contribution of this paper is the novel queuing model proposed to further improve task scheduling performance in cloud data centers. The steps to derive the joint probability distribution based on the proposed CQD model are given in detail. The model is further compared with other disciplines such as FIFO and priority queuing disciplines to reveal the improved performance by adopting CQD. It is rarely found that the probability distribution measures are computed for such real-time complex queuing systems. [2]

## 2 Literature Study

Few of the well-known service disciplines in queuing models are *First-In First-Out (FIFO)*, *Last-In First-Out (LIFO)*, *Selection In Random Order (SIRO)*, *Priority*, *Shortest Job First (SJF)*, and *Shortest Remaining Processing Time (SRPT)*. In cloud systems, priority of jobs, job lengths, job interdependencies, processing capacity and the current load on the VMs are the factors on deciding the next task to be scheduled [3], whereas above disciplines do not consider these.

In [4], the authors modeled the cloud center as a classic open network; they obtained the distribution of response time based on assumption that exponential inter-arrival time and service time distributions occurred. The response time distribution revealed the relationship between maximal number of tasks and minimal resource requirements against the required level of service. Generally, the literature in this area elaborately discusses M/G/m queuing systems, as outlined in [5–7]. The response time distributions and queue length analysis for M/G/m systems are insufficient for cloud environment as it requires two stages of processing where workload characterization and characterized task allocations are a must. As solutions for distribution of response time and queue length in M/G/m systems cannot be obtained in closed form, suitable approximations were sought in these papers.

However, most of the above models lead to proper estimates of mean response time with constraint that lesser VMs are present [8]. Approximation errors are particularly large when the offered load is small and the number of servers  $m$  are more [9]. Hence, these results are not directly applicable to performance analysis of cloud computing environments where generally number of VMs are large and service distributions and arrival distributions are not known.

### 3 Clustered Queuing Model

As observed, the existing queuing models do not directly fit in cloud environment [10]. Hence, a suitable queuing model that best depicts the cloud scenario is developed. In  $M/G/\infty$  networks, the analysis of waiting time and response time distributions is already known and well established, but the determination of the joint distribution of the queue lengths at the various servers at the arrival epochs of a submitted task in those nodes presents an important problem. This paper is devoted to this problem. The following subsection discusses the proposed queuing model in detail.

#### 3.1 Statistical Model

The focus is to derive a queuing model with the above characteristics in order to effectively schedule incoming task requests in cloud environment. Jockeying is allowed when load imbalance among VMs becomes high. Here, balking or renegeing scenarios are not considered to maintain simplicity.

The queuing model as shown in Fig. 1 involves a global and several local queues. The global queue is one-way entry into the system, and all the tasks submitted to the cloud environment pass through this queue. Arrival rate of incoming requests is taken as  $\lambda$ , and  $\mu_1$  and  $\mu_2$  are the clustering and scheduling service rates, respectively. Considering that departure process of each queue is again a Poisson process, the following discussion is put forth.

Clustering mechanism is based on each task's resource requirement rate. Workload characterization of incoming task requests is done here. Task requests are buffered for  $\Delta t$  units of time, and these task requests are clustered according to their resource requirement. Considering the cloud environment, the queuing model best suited is proposed. The queuing model prescribed for this problem is given in Kendall's notation as,

$$(M/G_2/m):(\infty/CQD)$$

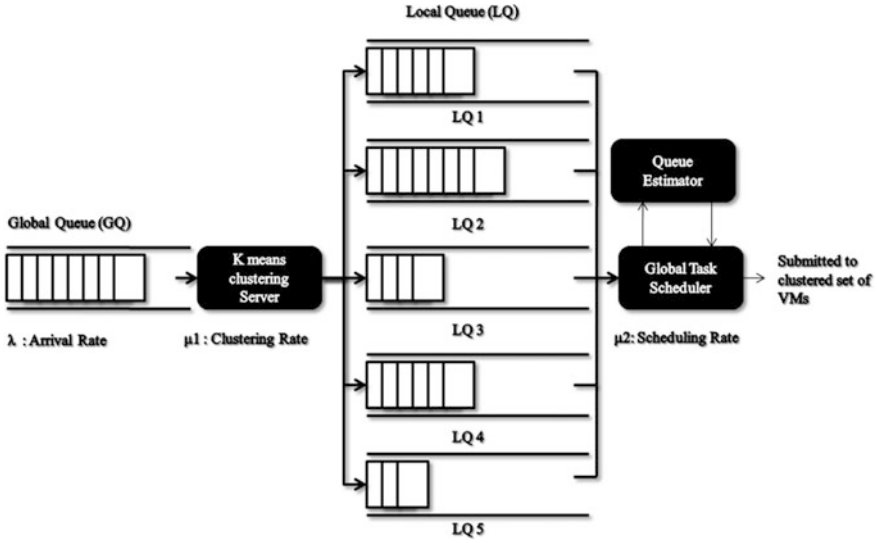


Fig. 1 Illustration of the queuing model

It represents that arrival follows Markovian arrival distribution (M) and the service follows two-stage general service distribution (G2) with  $m$  number of VMs in an infinite capacity system ( $\infty$ ). The model follows clustered queue discipline (CQD) as the queue discipline which is the proposed work. Here, general distribution means an arbitrary distribution with  $E(x)$  and  $E(x^2)$  and the service times are independent and identically distributed (IID) [11].

### 3.2 Performance Parameters

Generally, performance parameters involve terms such as server utilization, throughput, waiting time, response time, queue length, and number of customers in the system at any given time [2]. Here,  $\lambda$  is the arrival rate;  $\mu_1$  and  $\mu_2$  are clustering and scheduling service rates.

Server utilization factor for clustering server  $U_c$  with Poisson arrival and general service distribution is given by,

$$U_c = \lambda / \mu_1 \tag{1}$$

Server utilization factor scheduling server  $U_s$  with Poisson arrival and general service distribution with  $m$  number of VMs in the system is given by,

$$U_s = \lambda / m\mu_2 \tag{2}$$

Throughput of the system is defined as the mean number of requests serviced during a time unit.  $\lambda_{(i)}$  denotes the arrival of tasks at the parallel queues in stage II at queues  $i$  ranging from 1 to  $m$ . It is denoted using  $\psi$  and is given by,

$$\psi = U_c\mu_1 + U_s\mu_2 \left( \sum \lambda_{(i)} / m\mu_{2(i)} \right) \tag{3}$$

### 3.3 Joint Probability Distribution for Tandem Queues in Parallel

We approach the model as two queuing phases in series. The first phase is considered to follow single-server single-queue model with infinite service capacity, whereas the second phase involves tandem queues in parallel with single global scheduler as the server with infinite capacity. As the first part is a well-known single-server single-queue model, it does not require any further investigation. The second phase of the model with tandem queues in parallel is of major concern.

Some notations used to model the queuing discipline: [12].

$LQ_1, LQ_2, \dots, LQ_k$  are  $K$  queues in parallel and the tasks arrive at the queues in Poisson fashion with  $\lambda$  as arrival rate and service times at  $LQ_1, LQ_2, \dots, LQ_k$  are independent, identically distributed stochastic variables with distribution  $B_1(\cdot), B_2(\cdot), \dots, B_k(\cdot)$  with first moment  $\beta_1, \beta_2, \dots, \beta_k$ . In the following, it will be assumed that  $B_1(0+) = 0$  and  $\beta_1 < \infty, i = 1, \dots, k$ .

On deriving the queue characteristics of the second phase of the model, we shall compound the results with the already known parameters of  $M/M/1$  model [13].

Now, the derivation steps for tandem queues in parallel are discussed below. Our approach initially considers 2 queues in parallel and then extends the results to  $k$  queues in parallel.

The proposed model is derived from supplanting tandem queues in series derivation [1] into queues in parallel. The following three steps outline the derivation method adopted from the approach followed by O.J. Boxma:

- Determine a product-form expression of the type determining joint stationary queue length distribution of a submitted task at its arrival epochs at two queues of a general network of  $M/G/\infty$  queues;
- Apply the PASTA property which states that ‘Poisson Arrivals See Time Averages’ [14];
- Decompose the queue length term  $X_m(t)$  into independent terms corresponding to the position of a task at time instant 0.

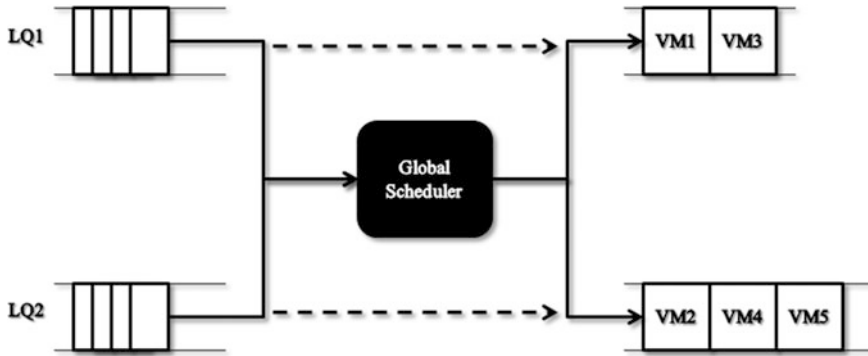


Fig. 2 2—M/G/m queues in parallel

In Fig. 2, let  $x_1(t)$  and  $x_2(t)$  denote the queue length of LQ1 and LQ2 at time  $t$  such that  $x_1(t) = l_1$  and  $x_2(t) = l_2$ . Let  $\sigma_1^{(1)}, \sigma_2^{(1)}, \dots, \sigma_{l_1}^{(1)}, \sigma_1^{(2)}, \sigma_2^{(2)}, \dots, \sigma_{l_2}^{(2)}$  denote residual service times of the tasks in service, i.e., remaining service time required by each task to complete. Hence,  $(x_1(t), x_2(t), \sigma_1^{(1)}, \sigma_2^{(1)}, \dots, \sigma_{l_1}^{(1)}, \sigma_1^{(2)}, \sigma_2^{(2)}, \dots, \sigma_{l_2}^{(2)})$  is evidently a Markov process.

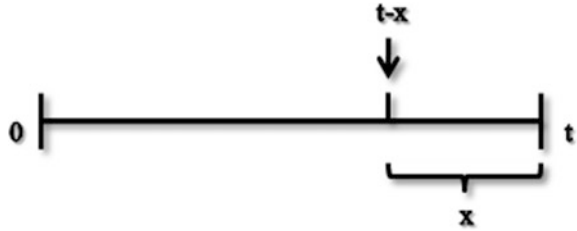
**Theorem** *At equilibrium, the joint stationary distribution of queue lengths  $l_1$  and  $l_2$  when a particular task arrives at either  $LQ_1$  or  $LQ_2$  at its arrival epoch is given by Eq. 4*

$$\begin{aligned}
 & \Pr\{x_1(t) = l_1, x_2(t) = l_2, \sigma_1^{(1)} \leq x_1, \sigma_2^{(1)} \leq x_2, \dots, \sigma_{l_1}^{(1)} \leq x_{l_1}, \sigma_1^{(2)} \leq y_1, \sigma_2^{(2)} \leq y_2, \dots, \sigma_{l_2}^{(2)} \leq y_{l_2} / x_1(0) = 0, x_2(0) = 0\} \\
 & = \exp\left(-\lambda \int_0^{l_1} (1 - B_1(x) * B_2(x)) \cdot dx\right. \\
 & \quad \left. + \frac{\lambda^{l_1}}{l_1!} \prod_{i=0}^{l_1} \left\{ \int_0^t (B_1(x + x_i) - B_1(x)) \cdot dx \right\} \right. \\
 & \quad \left. + \frac{\lambda^{l_2}}{l_2!} \prod_{j=1}^{l_2} \left\{ \int_0^t (B_2(x + y_j) - B_2(x)) \cdot dx \right\} \right), t = 0, l_1, l_2 = 0
 \end{aligned} \tag{4}$$

*Proof* Assuming that in the interval  $(0, t)$ ,  $n$  tasks arrive where  $n \geq l_1 + l_2$ . It is trivial that in a Poisson process of arrival between  $(0, t)$  the joint probability distribution of the epochs of these arrivals agrees with the joint distribution of  $n$  independent random points distributed uniformly in  $(0, t)$ .

As shown in Fig. 3, if a task arrives at epoch  $(t-x)$  then the task is clustered into either  $LQ_1$  or  $LQ_2$ . Then,  $B_1(x + x_i) - B_1(x)$  will be the distribution at  $LQ_1$  with

Fig. 3 Timeframe  $\Delta t$



residual service time at most  $x_i$ .  $(B_2(x + y_j) - B_2(x))$  will be the distribution at  $LQ_2$  with residual service time at most  $y_j$ . If the task has left the local queue, then the distribution will be  $B_1(x) * B_2(x)$ . Now, LHS can be written as,

$$\begin{aligned}
 \text{LHS} = & \sum_{n=l_1+l_2}^x \left( e^{-\lambda t} \frac{(\lambda t)^n}{n!} \frac{n!}{l_1!l_2!(n-l_1-l_2)!} \prod_{i=1}^{l_1} \left\{ \frac{1}{t} \int_0^t (B_1(x+x_i) \right. \right. \\
 & - B_1(x)) \cdot dx \Big\} + \prod_{j=1}^{l_2} \left\{ \frac{1}{t} \int_0^x (B_2(x+y_j) - B_2(x)) \cdot dx \right\} \\
 & + \left. \prod_{k=1}^{n-l_1 \text{ or } n-l_2} \left\{ \frac{1}{t} \int_0^t (B_1(x) * B_2(x)) \cdot dx \right\} \right) \quad (5)
 \end{aligned}$$

If we put  $t \rightarrow \infty$ , we obtain Eq. 6 after integration rearrangements. The argument can be extended to generalize that limiting distribution is independent of the initial distribution.

If  $\beta_1 < \infty, \beta_2 < \infty$ , then,

$$\begin{aligned}
 \Pr\{x_1(t) = l_1, x_2(t) = l_2, \sigma_1^{(1)} \leq x_1, \sigma_2^{(1)} \leq x_2, \dots, \sigma_{l_1}^{(1)} \leq x_{l_1}, \sigma_1^{(2)} \leq y_1, \sigma_2^{(2)} \leq y_2, \dots, \sigma_{l_2}^{(2)} \leq y_{l_2}\} \\
 = e^{-\lambda \beta_1} \frac{(\lambda \beta_1)^{l_1}}{l_1!} \prod_{i=0}^{l_1} \left\{ \int_0^{x_i} \frac{1 - B_1(x)}{\beta_1} \cdot dx \right\} \\
 + e^{-\lambda \beta_2} \frac{(\lambda \beta_2)^{l_2}}{l_2!} \prod_{j=1}^{l_2} \left\{ \int_0^{y_j} \frac{1 - B_2(y)}{\beta_2} \cdot dy \right\} \quad (6)
 \end{aligned}$$

*Remarks* The above result is a limiting case of one of the models proposed by Cohen [15]. for processor sharing discipline. Equation 6 yields the well-known joint probability distribution of two stationary parallel queues. Now, on applying PASTA property, it follows that joint stationary distribution of queue lengths and residual service times just before the arrival of tagged task at LQs is given by the following. Then, the generating function of the joint stationary distribution of the queue lengths  $l_1$  and  $l_2$  is given by,



$$\begin{aligned}
& E [z_1^{l_1}] + E [z_2^{l_2}] \\
&= \int_{t=0}^{\infty} dB_1(t) \sum_{l_1=0}^{\infty} z_1^{l_1} \sum_{l_2=0}^{\infty} z_2^{l_2} \left\{ \int_{y_{n_2}=0}^{\infty} \Pr\{x_2(t) = l_2/x_1(0) = l_1, x_2(0) = n_2, \tau \right. \\
&= t, \sigma_1^{(1)} = x_1, \dots, \sigma_{l_1}^{(1)} = x_{l_1}, \sigma_1^{(2)} = y_1, \dots, \sigma_{n_2}^{(2)} \\
&= y_{n_2}\} e^{-\lambda\beta_1} \frac{(\lambda\beta_1)^{l_1}}{l_1!} \prod_{i=0}^{l_1} \left\{ \frac{1 - B_1(x_i)}{\beta_1} \right\} \\
&+ e^{-\lambda\beta_2} \frac{(\lambda\beta_2)^{n_2}}{l_2!} \prod_{j=1}^{n_2} \left\{ \frac{1 - B_2(y_j)}{\beta_2} \right\} \cdot dx_{l_1} \cdot dy_{n_2} \\
&+ \int_{x_{n_1}=0}^{\infty} \Pr\{x_1(t) = l_1 \text{ x hew 1 by the following proof } // /x_1(0) \\
&= n_1, x_2(0) = 0, \tau = t, \sigma_1^{(1)} = x_1, \dots, \sigma_{n_1}^{(1)} = x_{n_1}, \sigma_1^{(2)} = y_1, \dots, \sigma_{l_2}^{(2)} \\
&= y_{l_2}\} e^{-\lambda\beta_1} \frac{(\lambda\beta_1)^{n_1}}{n_1!} \prod_{i=0}^{n_1} \left\{ \frac{1 - B_1(x_i)}{\beta_1} \right\} \\
&+ e^{-\lambda\beta_2} \frac{(\lambda\beta_2)^{l_2}}{l_2!} \prod_{j=1}^{l_2} \left\{ \frac{1 - B_2(y_j)}{\beta_2} \right\} \cdot dx_{n_1} \cdot dy_{l_2} \left. \right\} \quad |z_1| \leq 1, |z_2| \leq 1
\end{aligned} \tag{7}$$

Now, independent terms for  $x_1(t)$  and  $x_2(t)$  for both the local queues are to be derived following term decomposition technique given by Cohen [15]. Combining the independent terms for both the queues gives the following equation.

$$\begin{aligned}
E [z_1^{l_1}] + E [z_2^{l_2}] &= e^{-\lambda\beta_1(z_1-1)} \int_0^{\infty} e^{-\lambda\beta_1 z_1 p_0(1-z_1)} \cdot dB_1(t) \\
&+ e^{-\lambda\beta_2(z_2-1)} \int_0^{\infty} e^{-\lambda\beta_2(1-p_2)} \cdot dB_2(t) \quad |z_1| \leq 1, |z_2| \leq 1
\end{aligned} \tag{8}$$

The above equation gives the joint stationary distribution of the two local queues considered. On extending the above argument to  $n$  arbitrary local queues, we can arrive at the final joint distribution. The above scenario of  $n$  different classes of parallel queues is simulated, and analysis is given in the following section.

## 4 Experimental Analysis

The CQD policy is analyzed against other well-known policies such as priority and FIFO mechanisms. Here, M/G/m parallel queues are considered for experimentation. Existing literature [16, 17] deals with performance analysis of queuing systems based mainly on mean response time which is highly critical in cloud environment to provide necessary quality of service (QoS) [18].

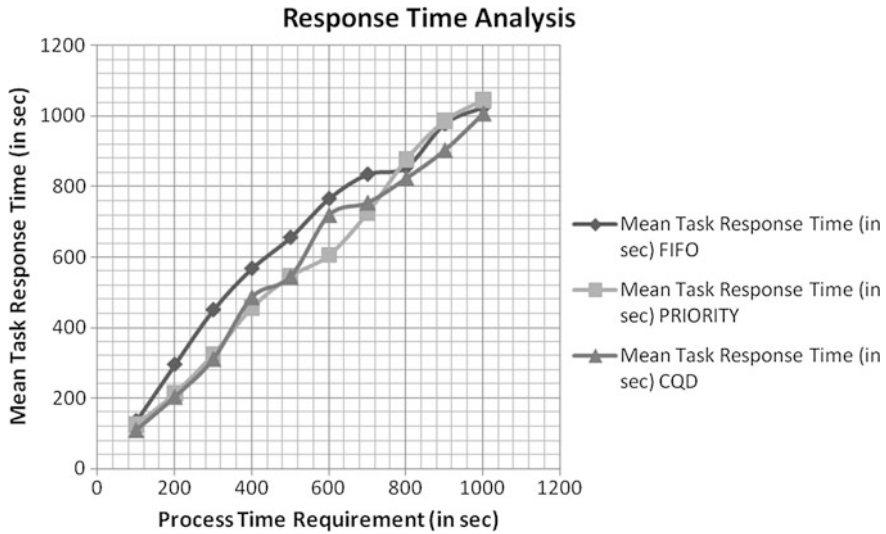


Fig. 4 Mean task response time analysis based on processing requirements

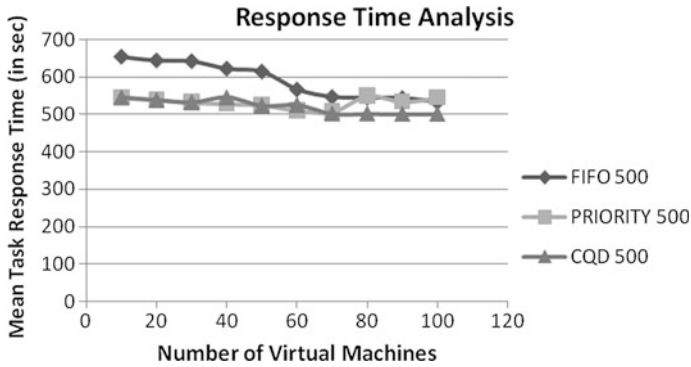
### 4.1 Performance Metrics

The mean task response time for tasks that are queued following FIFO, priority, and CQD disciplines is analyzed as follows: The number of VMs is fixed in this case as 10. For FIFO and priority disciplines, the mean response time function steeply increases as requirement increases. But CQD exhibits a smooth incremental change as the task requirement increases. This smooth increase is due to the clustering of appropriate class of tasks to appropriate VM, thereby reducing excess waiting time (Fig. 4).

The trend in Fig. 5 shows that, as the number of VMs increases, there is a sharp increase in CQD performance compared to FIFO and priority. Queuing based on priority and CQD is almost similar in performance as evident from Fig. 5. As the number of VMs increases, the mean task response time is found to decrease in both priority and CQD in a similar fashion. Hence, the model proves to work in par with priority queue discipline for high number of VMs and lesser number of task requests. But as the task requests increases, CQD significantly outperforms priority and FIFO queue mechanisms.

### 4.2 Analysis and Discussion

The space complexity depends on the buffer space in each and every VM. It has to be effectively planned to use optimum buffer space. The average waiting time of a task [19] in this type of queuing system will comprise of waiting time in stage 1 and



**Fig. 5** Mean task response time analysis based on number of VMs with number of tasks fixed as 500

in stage 2, time order of clustering algorithm and time order of scheduling policy adopted by the global scheduler. It is given as,

$$WT_i = W_{s1} + O(\text{clustering}) + W_{s2} + O(\text{scheduling}) \quad (4)$$

**Amortized analysis.** Not each of the  $n$  tasks takes equally much time. Basic idea in CQM is to do a lot of ‘prework’ by clustering a-priori. This pays off as a result of the prework done, and the scheduling operation can be carried out so fast that a total time of  $O(g(n))$  is not exceeded where  $g(n)$  is a sub-linear function of  $n$  tasks. So, the investment in the prework amortizes itself.

## 5 Conclusion and Future Work

This paper outlines the need for efficient queuing model which is best suited for cloud computing. A novel method involving clustering technique is proposed. The queuing model derivation steps are outlined and validated against existing queues in series derivation. Analytical discussion proves the efficiency of the above method. The proposed work is found to perform better than existing disciplines such as FIFO and priority in situations such as high resource requirement and when large number of VMs are present. Major work in the future shall be devoted to applying the model in real time and mathematically deriving and analyzing the efficiency in terms of energy complexity.

**Acknowledgements** We acknowledge Visvesvaraya PhD scheme for Electronics and IT, DeitY, Ministry of Communications and IT, Government of India’s fellowship grant through Anna University, Chennai for their support throughout the working of this paper.

## References

1. Boxma, O.J.:  $M/G/\infty$  tandem queues. *Stoch. Process. Appl.* **18**, 153–164 (1984)
2. Sztrik, J.: *Basic Queueing Theory*. University of Debrecen (2012)
3. Buyya, R., Sukumar, K.: *Platforms for Building and Deploying Applications for Cloud Computing*, pp. 6–11. CSI Communication (2011)
4. Xiong, K., Perros, H.: Service performance and analysis in cloud computing. In: *Proceedings of the 2009 Congress on Services—I*, Los Alamitos, CA, USA, pp. 693–700 (2009)
5. Ma, B.N.W.: Mark. J.W.: Approximation of the mean queue length of an  $M/G/c$  queueing system. *Oper. Res.* **43**, 158–165 (1998)
6. Miyazawa, M.: Approximation of the queue-length distribution of an  $M/GI/s$  queue by the basic equations. *J. Appl. Probab.* **23**, pp. 443–458 (1986)
7. Yao, D.D.: Refining the diffusion approximation for the  $M/G/m$  queue. *Oper. Res.* **33**, 1266–1277 (1985)
8. Tijms, H.C., Hoorn, M.H.V., Federgru, A.: Approximations for the steady-state probabilities in the  $M = G=c$  queue. *Adv. Appl. Probab.* **13**, 186–206 (1981)
9. Kimura, T.: Diffusion approximation for an  $M = G=m$  queue. *Oper. Res.* **31**, 304–321 (1983)
10. Vilaplana, Jordi, Solsona, Francesc, Teixidó, Ivan, Mateo, Jordi, Abella, Francesc, Rius, Josep: A queuing theory model for cloud computing. *J Supercomput.* **69**(1), 492–507 (2014)
11. Boxma, O.J., Cohen, J.W., Huffel, N.: Approximations of the Mean waiting time in an  $M = G=s$  queueing system. *Oper. Res.* **27**, 1115–1127 (1979)
12. Kleinrock, L.: *Queueing Systems: Theory*, vol. 1. Wiley-Interscience, New York (1975)
13. Adan, I.J.B.F., Boxma, O.J., Resing, J.A.C.: Queueing models with multiple waiting lines. *Queueing Syst Theory Appl* **37**(1), 65–98 (2011)
14. Wolff, R.W.: Poisson arrivals see time averages. *Oper. Res.* **30**, 223–231 (1982)
15. Cohen, J.W.: The multiple phase service network with generalized processor sharing. *Acta Informatica* **12**, 245–284 (1979)
16. Khazaei, H., Mistic, J., Mistic, V.: Performance analysis of cloud computing centers using  $M/G/m/m + r$ . *Queueing Systems. IEEE Trans. Parallel Distrib. Syst.* **23**(5) (2012)
17. Slothouber, L.: A model of web server performance. In: *Proceedings of the Fifth International World Wide Web Conference* (1996)
18. Yang, B., Tan, F., Dai, Y., Guo, S.: Performance evaluation of cloud service considering fault recovery. In: *Proceedings of the First International Conference on Cloud, Computing (CloudCom'09)*, pp. 571–576 (2009)
19. Borst, S., Boxma, O.J., Hegde, N.: Sojourn times in finite-capacity processor-sharing queues. *Next Gener. Internet Netw IEEE* 55–60 (2005)

# Static and Dynamic Analysis for Android Malware Detection



Krishna Sugunan, T. Gireesh Kumar and K. A. Dhanya

**Abstract** In this work, we perform a comparative study on the behavior of malware and benign applications using its static and dynamic features. In static analysis, the permissions required for an application are considered. But in dynamic, we use a tool called Droidbox. Droidbox is an android sandbox which can monitor some app actions like network activities, file system activities, cryptographic activities, information leakage, etc. Here, we consider these actions as well as dynamic API calls of applications. We propose to implement an android malware detector that can detect an app whether it is malware or not, prior to installation.

**Keywords** Malware · Benign · Static · Dynamic · Droidbox

## 1 Introduction

The spread of mobile devices is uncontrollable because they bring new changes to everyday life. Android smartphone Operating System has captured above 75% of the entire market share. Android market such as Google playstore and other third party store plays a major part in the familiarity of android devices. However, the open nature of android generates these markets as the hot spot for malware offensives and creates infinite occurrences of malware being masked in a wide number of benign apps that vigorously intimidate the end user's security and privacy. There are variety of android malwares such as premium rate SMSTrojan, aggressive adware, spyware, botnet, and privilege escalation exploits [1] which are distributed through third party and official app store. There are two ways for analyzing the malware, static approach and dynamic approach [2]. Static method is just reading or disassemble the

---

K. Sugunan (✉) · T. Gireesh Kumar · K. A. Dhanya  
TIFAC-CORE in Cyber Security, Amrita School of Engineering,  
Amrita Vishwa Vidyapeetham, Amrita University, Coimbatore, India  
e-mail: krishnasugunan256@gmail.com

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_13](https://doi.org/10.1007/978-981-10-7200-0_13)

application without running. The signature-based approach is an example of static analysis, in which it compares the signature of application with the already existing signature database, it is predefined. It becomes ineffective when the new variant of malware is formed. While in dynamic analysis, the app is executed in a protected environment which is based on the behavioral aspects.

In this proposed work, the features from static analysis and dynamic analysis of android applications are collected. Static features are considered as permissions extracted from the manifest file, while dynamic analysis is based on some app actions like information leakage, network activities, file activities, cryptographic operations, Short Message Services, phone calls, and API calls at run time. Then, a comparative study on the behavior of both benign and malware applications is performed. It helps to identify even if an app is malware or not.

The remaining sections are as follows. Section 2 depicts the related works. Section 3 illustrates the proposed work. Section 4 explains result obtained from experiment and finally Sect. 5 elucidates conclusion.

## 2 Related Works

### 2.1 *DroidDetector: Android Malware Characterization and Detection Using Deep Learning*

Yuan et al. [3] proposed to collaborate the features from static and dynamic analysis of android apps. In static analysis, they considered both permissions and sensitive API calls of application. They looked for 120 permissions and 59 sensitive API calls. While in dynamic analysis; they used a tool called Droidbox which is an android sandbox, they monitored 13 app actions from it. They combined both static and dynamic features, the total of 192 features. Then, they created the binary feature vector depending on the existence and nonexistence of particular feature in the application. Then, this collected feature goes to deep learning model for detection.

### 2.2 *Merging Permission and API Features for Android Malware Detection*

Qiao et al. [4] suggested a system, in which different machine learning method to identify malware by using the arrangement of permissions and API function calls in the android application. They defined the collection of all permissions as P and collection of functions as F. After that, they defined the association map between functions and permissions. Then, they computed the binary API features, numerical

API features, binary permission features, and numerical permission features. After that, they performed data analysis. The permission set included 104 binary features and numerical features, while the API contained 654 binary features and numerical features. Then, they applied some machine learning techniques such as RF, SVM, and ANN in both binary and numerical feature set. And they found that the numerical feature set shows more stable performance than binary feature set.

### ***2.3 DroidMat: Android Malware Detection Through Manifest and API Calls Tracing***

Wu et al. 2012 [5] proposed static feature-based approach to find out whether the android application is infected or not. Here, they considered static information containing permissions, distribution of components, intent messages passing, and API calls for identifying android malware application nature. They used k-means algorithm for clustering and selected the best clusters from it using single value decomposition algorithm. Then, it goes for malware identification phase. It had many disadvantages such as it cannot perform well on detection of basebridge and droid-kunfu android malware.

## **3 Proposed Work**

For the effective analysis of android apps, perform static as well as dynamic analysis to get features from each application. All the collected features from each app fall under the group of permissions, dynamic behaviors, and API calls. Permissions are extracted through static analysis while API calls and dynamic behaviors are collected through dynamic analysis.

### ***3.1 Dataset***

We collected 150 malware applications from the Drebin dataset [6]. For benign applications, we randomly downloaded 200 applications from Google playstore and verified them in virusTotal.

### 3.2 Extraction of Permission Features

In static analysis phase, we use a reverse engineering tool called APKtool [7] for uncompressing the apk file and read the permissions from manifest file using AXML-Printer2 [8]. Figure 1 represents the flowchart of permission extraction. In this step, we consider a total of 94 android permissions. Then, binary permission feature vectors are generated. If the permission exists in each app, then it is marked as 1 otherwise 0. For better understanding the permissions that are used in benign as well as malware applications, take the frequency of each permissions and represent it in a bar diagram. In Fig. 2, x-axis shows the permissions used and y-axis represents the frequency of occurrence.

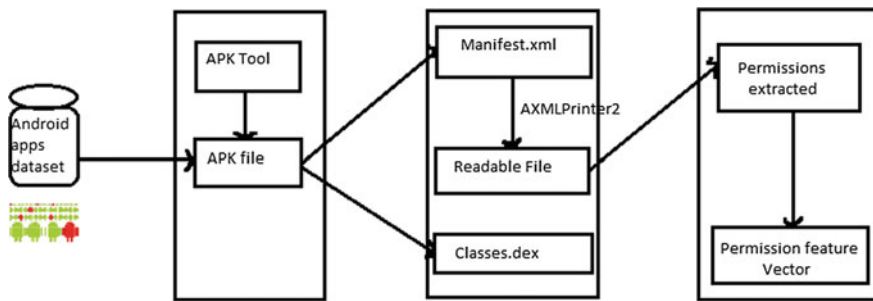


Fig. 1 Flowchart of permission extraction

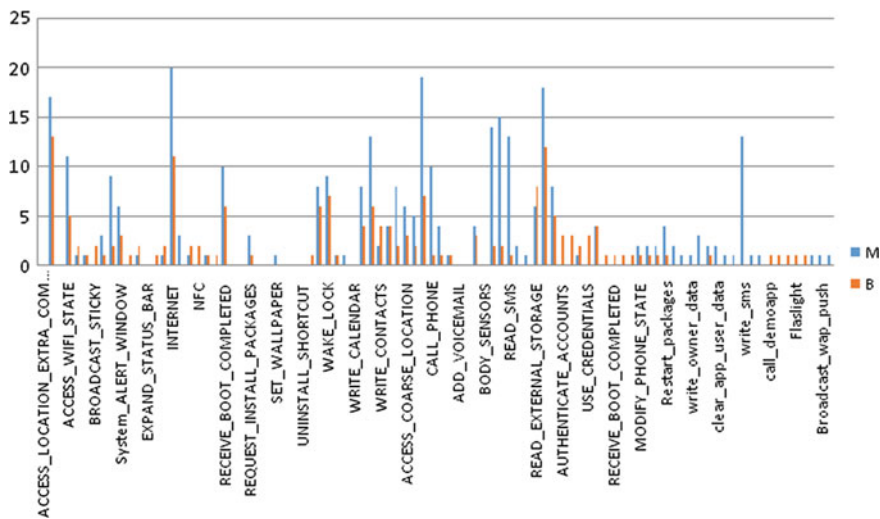


Fig. 2 Android application permissions

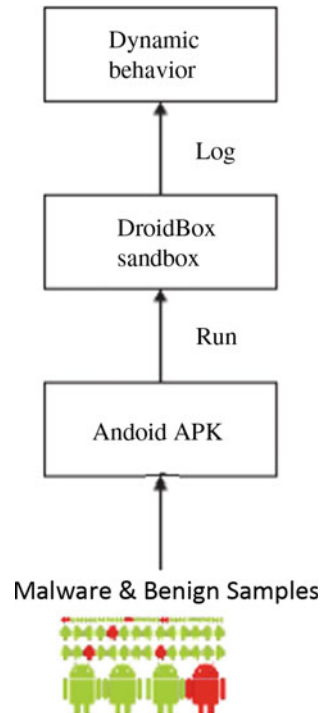


### 3.3 Extraction of Dynamic Behaviors

This is a dynamic approach in which, a separate environment for installing and running the application is used. And some features are collected from it. Here, we use a tool called Droidbox [8], which is an android sandbox. It monitors some actions of applications such as file read and write operations, incoming/outgoing network data, cryptographic operations, information leaks, send SMS and phonecalls. It's a very good tool for analyzing and understanding the behavior of application during run time. In this work, app is ran in the Droidbox for 60 s time and collected the executed app actions from it. In this phase, we observed 13 actions from it. Figure 3 represents the flowchart for the extraction of dynamic behaviors.

Then, the binary feature vector is generated, which is based on the presence or absence of particular actions in each app. If the action persists, then it is taken as 1 otherwise 0. For better understanding of dynamic actions, the frequency of each action in the application is taken and is plotted using x and y coordinates. In Fig. 4, x-axis depicts the app actions and the y-axis illustrates the frequency.

**Fig. 3** Extraction of dynamic behavior



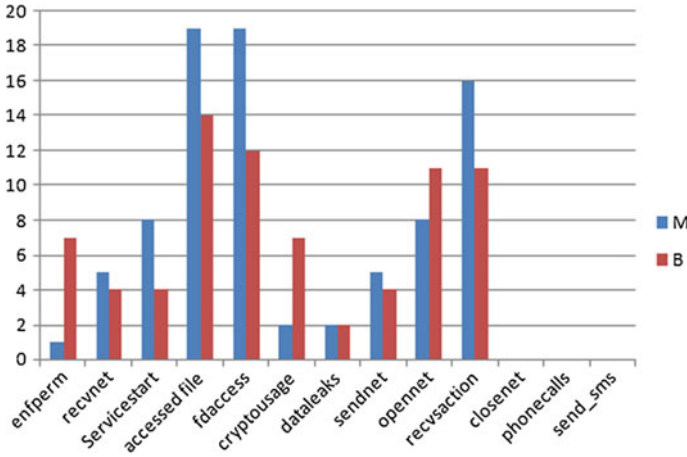
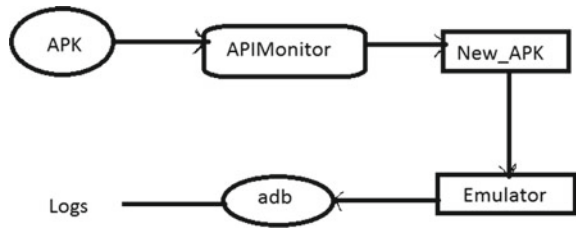


Fig. 4 Application activities

Fig. 5 Extraction of API calls



### 3.4 Extraction of API Calls

It is not enough for a good dynamic analysis of mobile malware. So we analyzed the dynamic API call logs of the application along with these dynamic actions. API call logs help to understand the behavior of applications. A tool called APIMonitor [9] is used for analyzing the APICalls in the application. Firstly, the apk file is submitted in APIMonitor tool, and a repackaged apk is formed with new apk tag. Then, this new apk is installed and ran in emulator and collected the API call logs using adb logcat command. Here, we considered 60 API calls and generated the numerical API feature vector. This feature set contains the count of each API calls in each applications. Figure 5 represents the extraction of API calls. In Fig. 6, x-axis represents the API calls and y-axis represents the frequency of API calls in benign and malware applications. Three API calls have large frequency compared to others so it is plotted separately.

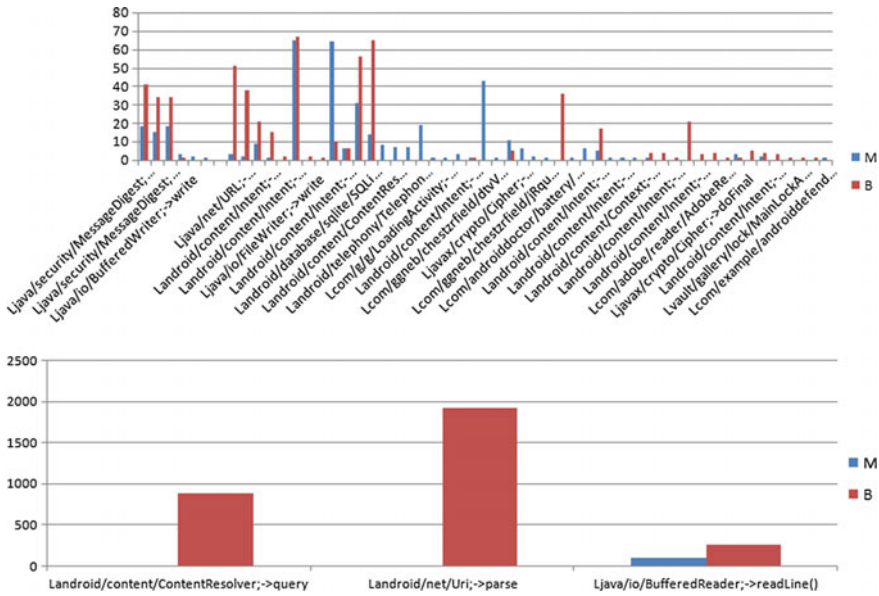


Fig. 6 API calls

### 4 Results and Discussion

Both benign and malicious applications are distinguished on the basis of the extracted features. Our work consists of two binary datasets and one numerical dataset. The binary dataset represents the presence and absence of permissions and some behavioral actions. And numerical dataset represents the frequency of each API calls. Table 1 lists the Weka [10] classification results from the analysis with binary permission features, binary dynamic behavior features, and numerical API call features independently without feature selection using Naive Bayes [11], Random Forest [12], SVM [13], and J48 [14]. Then, it is compared with combined features. From the above comparison, we found that the combination is effective than individual. Then, some top permissions and API calls features are selected based on its frequency of occurrence in benign and malicious applications. For example, some permissions in permission feature set like read\_contacts, camera, etc. have a high frequency of occurrence in malware applications than benign. Then, classification algorithms like Naive Bayes, SVM, RF, and J48 are applied in this selected feature set using Weka tool. Table 2 shows its classification results. From the above comparison, combination of selected features is better than previous one.

**Table 1** Classification results without feature selection

		Precision	Recall	F-measure
Permission	Naive Bayes	0.905	0.857	0.863
	SVM	0.857	0.714	0.726
	RF	0.857	0.714	0.726
	J48	0.857	0.714	0.726
Dynamic behaviors	Naive Bayes	0.857	0.714	0.726
	SVM	0.905	0.857	0.863
	RF	0.905	0.857	0.863
	J48	0.905	0.857	0.863
API	Naive Bayes	0.920	0.909	0.906
	SVM	0.805	0.788	0.774
	RF	0.748	0.727	0.732
	J48	0.748	0.727	0.732
Combined features	Naive Bayes	0.905	0.857	0.863
	SVM	0.720	0.727	0.717
	RF	0.91	0.86	0.87
	J48	0.857	0.714	0.726

**Table 2** Classification results after feature selection

		Precision	Recall	F-measure
Permission	Naive Bayes	0.905	0.857	0.863
	SVM	0.905	0.857	0.863
	RF	0.905	0.857	0.863
	J48	0.857	0.714	0.726
Numeric API	Naive Bayes	0.857	0.714	0.726
	SVM	0.905	0.857	0.863
	RF	0.748	0.727	0.732
	J48	0.829	0.727	0.732
Combined features	Naive Bayes	0.857	0.714	0.726
	SVM	0.905	0.857	0.863
	RF	0.905	0.857	0.863
	J48	0.905	0.857	0.863

## 5 Conclusion and Future Works

The fundamental objective of this work is to analyze the behavior of benign as well as malicious android apps based on the number of features collected from the static along with dynamic analysis. Almost all the malware detection system uses either static or dynamic analysis. Here, we proved that the combination of features from static and dynamic analysis is more effective than the separate feature using some machine learning algorithms like RF, SVM, J48, and Naive Bayes. In future, we are planning to implement a deep learning model that can automatically identify whether an android app is malware infected or not prior to installation. Deep learning is a advanced field of machine learning which is a part of artificial intelligence. It mimics the manner the human brain works, which is more effective than other machine learning techniques. The feature vector obtained from the analysis of application samples is given as the input to deep learning model. The feature selection and reductions are performed within this model. Deep learning model consists of numerous invisible layers. If the number of invisible layer is increased, the performance becomes effective.

## References

1. Feizollah, A., Anuar, N.B., Salleh, R., Wahab, A.W.A.: A review on feature selection in mobile malware detection. *Digit. Invest.* **13**, 22–37 (2015)
2. Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M.S., Conti, M., Rajarajan, M.: Android security: a survey of issues, malware penetration, and defenses. *IEEE Commun. Surv. Tutor.* **17**(2), 998–1022 (2015)
3. Yuan, Z., Lu, Y., Xue, Y.: Droiddetector: android malware characterization and detection using deep learning. *Tsinghua Sci. Technol.* **21**(1), 114–123 (2016)
4. Qiao, M., Sung, A.H., Liu, Q.: Merging permission and api features for android malware detection. In: 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAIAAI), pp. 566–571. IEEE, (2016)
5. Wu, D.-J., Mao, C.-H., Wei, T.-E., Lee, H.-M., Wu, K.-P.: Droidmat: android malware detection through manifest and api calls tracing. In: 2012 7th Asia Joint Conference on Information Security (Asia JCIS), pp. 62–69. IEEE, (2012)
6. <https://www.sec.cs.tu-bs.de/~danarp/drebin/>
7. <https://ibotpeaches.github.io/Apktool/>
8. <http://android.amberfog.com/?tag=axmlprinter2>
9. <http://blog.dornea.nu/2014/08/05/android-dynamic-code-analysis-mastering-droidbox/>
10. <https://github.com/pjlantz/droidbox/wiki/APIMonitor>
11. <http://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/>
12. [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)
13. [https://en.wikipedia.org/wiki/Random\\_forest](https://en.wikipedia.org/wiki/Random_forest)
14. [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine)
15. <http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree>

# Performance Analysis of Statistical-Based Pixel Purity Index Algorithms for Endmember Extraction in Hyperspectral Imagery



S. Graceline Jasmine and V. Pattabiraman

**Abstract** This paper presents two endmember extraction (EE) algorithms which are based on single skewer and multiple skewers, respectively, to identify the pixel purity index (PPI) in hyperspectral images. In the existing PPI algorithm, the skewers were generated randomly which can generate dissimilar results in each of the iterations, and therefore, it may lead to the increase of false alarm probability. This issue has been resolved in these EE algorithms by generating skewers using statistical parameters of the hyperspectral dataset. This reduces the false alarm probability as well as the computational complexity of the conventional PPI algorithm. This work has been experimented using cuprite dataset. Experimental results prove the effectiveness of these EE algorithms in better identification of pure pixels.

**Keywords** Endmember · Hyperspectral image · Pixel purity index  
Skewer · Pure pixel

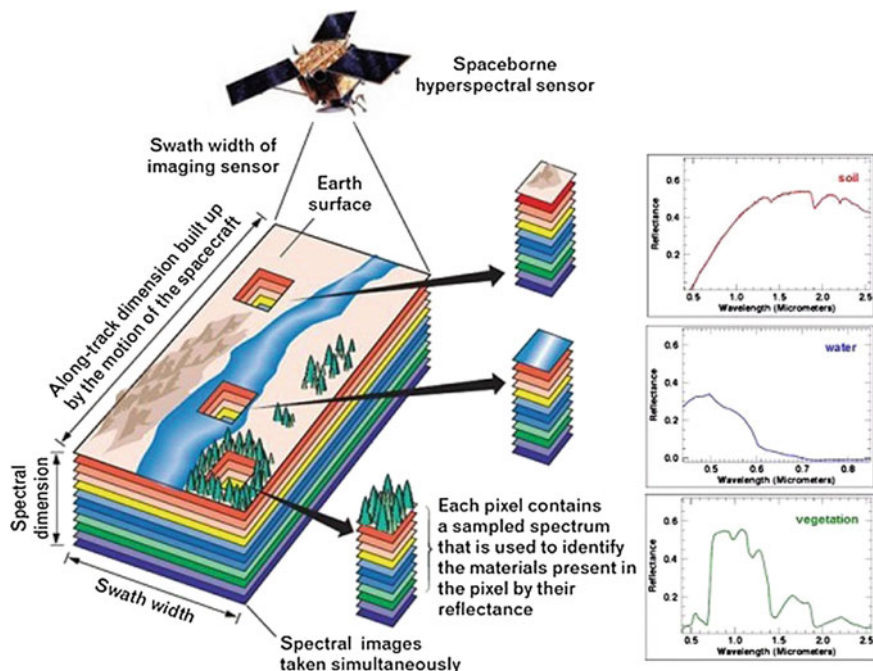
## 1 Introduction

Hyperspectral imaging (HSI) sensors are used to capture images of different wavelength channels, for the same area on the surface of the Earth [1]. The concept of HSI was devised at NASA's Jet Propulsion Laboratory in Pasadena, CA, USA, which developed instruments such as the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [2]. This instrument is capable to record the visible and

---

S. Graceline Jasmine (✉) · V. Pattabiraman  
School of Computing Science and Engineering, VIT University - Chennai Campus,  
Chennai, India  
e-mail: graceline.jasmine@vit.ac.in

V. Pattabiraman  
e-mail: pattabiraman.v@vit.ac.in



**Fig. 1** Concept of hyperspectral imaging

near-infrared spectrum of the reflected light of an area 2–12 km and several kilometres long using 224 spectral bands [3]. The hyperspectral data recorded by AVIRIS will be a collection of images which can be viewed as a three-dimensional cube, where the third dimension is the number of bands [4] and each vector corresponds to an individual pixel which is having a unique spectral signature to identify the objects in that spatial location. This is well demonstrated in Fig. 1. These images are discrete and are captured within the wavelength ranging from 0.4 to 2.5  $\mu\text{m}$ .

Hyperspectral imaging (HSI) focuses on analysing and interpreting the spectra of a particular scene acquired by a hyperspectral sensor. The spatial resolution of the sensor determines the precision of an image which can be further used to identify the materials available in the image. The low spatial resolution in HSI leads to a problem known as mixed pixel. Mixed pixel is a pixel, where many materials are observed through a single pixel. Mixed pixels have to be further classified to process any of the applications. Spectral unmixing is the process of decomposing the mixed pixel into a set of pure spectral signatures which can also be called as endmembers [5]. The mixed pixels are further classified into a set of classes, where

each class corresponds to a particular endmember and its abundance. Abundances indicate the proportion of each endmember present in the pixel. But high spatial resolution in HSI produces images containing both pure and mixed pixels.

## 2 Endmember Exaction Process

Endmember extraction is the major process involved in spectral unmixing. This step is required to find the set of pure spectral signatures from the hyperspectral scene under consideration. In the past decade, many algorithms were proposed related to endmember extraction. The research moves towards two different assumptions. The first assumption is that in the dataset, there will be at least one pure pixel for each endmember. A pixel can be called as pure pixel if that pixel is fully composed of one material. The algorithms which are based on pure pixel assumption are PPI [6], N-finder [7], vertex component analysis [8] and the maximum volume by householder transformation (MVHT) [9].

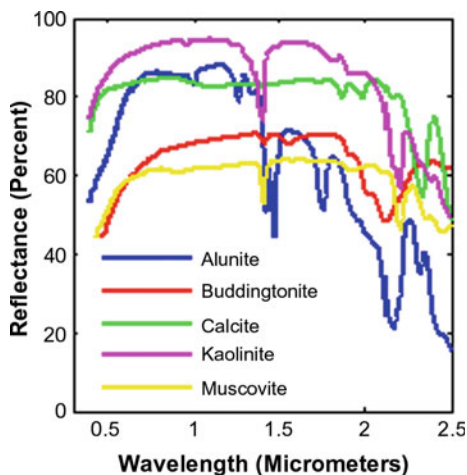
The second assumption is based on the absence of pure pixel. Some of the algorithms which are based on the absence of pure pixel assumption are the minimum volume simplex analysis (MVSA) [10], the simplex identification via split augmented Lagrangian (SISAL) [11] and convex cone analysis (CCA) [12]. In this paper, the performance analysis of statistical-based single skewer pixel purity algorithm (SBSS), statistical-based multiple skewer pixel purity algorithm (SBMS) and conventional pixel purity index algorithm (PPI) was done. SBSS and SBMS algorithms consider the first assumption of pure pixel existence in the hyperspectral image. The skewer generation process in the SBSS and SBMS algorithms has been done by considering some of the statistical parameters of the dataset. This produces better identification of pure pixels when compared to conventional PPI algorithm, and hence, it enhances the identification of endmembers in the dataset.

## 3 Hyperspectral Dataset

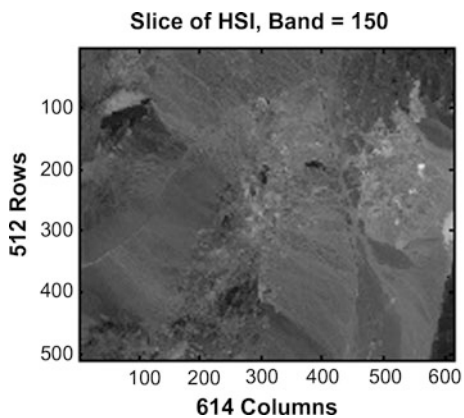
The dataset used for this experiment is AVIRIS cuprite scene, available online at the following website: [http://aviris.jpl.nasa.gov/data/free\\_data.html](http://aviris.jpl.nasa.gov/data/free_data.html). The scene selected for experimenting and processing in this paper is 970619t01p02\_r02\_sc04.a.rf. This scene has 224 spectral bands, 614 samples and 512 lines. From this range of spectral bands, the bands 1 to 3, 105 to 115 and 150 to 170 were removed due to water absorption and having low SNR value. So the total number of bands used for the further process has been reduced to 189. The minerals which are available in that area are alunite, buddingtonite, calcite, kaolinite and muscovite and are shown in Fig. 2.



**Fig. 2** Spectral signatures of the major material components found in cuprite scene



**Fig. 3** Cuprite scene at band = 150



The spectral signatures of these minerals were available in ASTER spectral library version 2.0 which was provided by California Institute of Technology. These signatures are used to compare and assess the endmember signatures in the scene processed [13]. Figure 2 shows the slice of cuprite hyperspectral image at band 150 Fig. 3.

## 4 Statistical-Based Skewer Generation Algorithms

SBSS and SBMS are the enhanced version of original PPI algorithm which improves the efficiency of the algorithm by eliminating the randomness in the process of generating the skewers. The skewers were used to identify skewness of the dataset.

### 4.1 Structure of the Input Hyperspectral Data for SBSS and SBMS

The entire hyperspectral dataset can be viewed as a three-dimensional data cube formed by a set of  $N$  discrete set of two-dimensional images. Each two-dimensional image is a composition of  $M$  pixels of size  $p \times q$ , and each pixel corresponds to the reflectance acquired for that spatial location. Each spectral vector,  $S_i$  corresponds to the reflectance of a particular pixel in all spectral bands, where  $S_i = [S_{i1}, \dots, S_{iM}]$ . Therefore, the hyperspectral data cubes will have extremely high dimension.

### 4.2 Steps for Skewer Generation in SBSS

1. Parameter  $\mu$  is used for the skewer generation, where  $\mu$  is a vector which contains the mean of the spectral values of each band.
2. The row size of the skewer, i.e. a one-dimensional vector  $\mu$ , is equal to the number of bands.
3. Finally, a single skewer,  $K$ , is generated using  $\mu$  as its value.

### 4.3 Steps for Skewer Generation in SBMS

1. Two parameters  $\mu$  and  $\alpha$  are used for the skewer generation, where  $\mu$  is a vector which contains the mean of the spectral values of each band and  $\alpha$  is a vector which contains the average of minimum and maximum spectral values of each band.
2. The row size of the single-dimensional vector  $\mu$  is equal to the number of bands, and the row size of the single-dimensional vector  $\alpha$  is equal to the number of bands.
3. Multiple skewers,  $K_i$ , are generated using  $\mu$  and  $\alpha$  as its boundary value.

#### 4.4 Steps for Data Projection in SBSS and SBMS

1. Let  $K$  be the number of skewers.
2. In all the iterations, the data points were projected upon these skewers.
  - 2.1 Projection is done by a dot operation

$$P = (p \times q) \cdot (K)$$

The result of the dot operation will be a vector,  $P$ , of size  $(p \times q)$ .

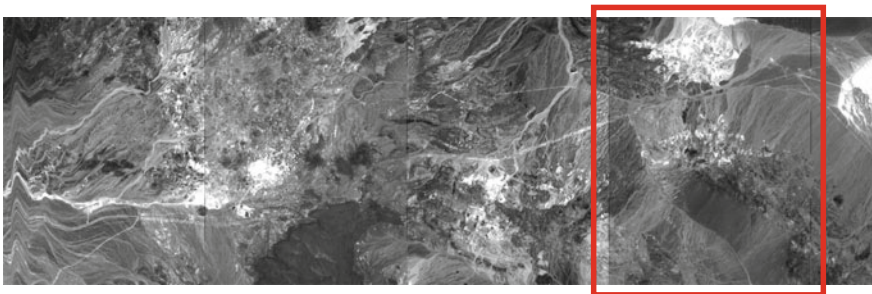
- 2.2 Choose the maximum value from the vector  $P$ , and note its index position as  $P_i$ .

#### 4.5 Steps for Purity Score Calculation

1. Initialise a count variable and mark zero for all pixels.
2. Whenever a pixel's index position is chosen as maximum in Step 2.2 of data projection, then the count for that pixel is incremented by 1.
3. The pixels whose count is marked more than zero are chosen as pure pixels.

## 5 Results and Discussion

The SBSS and SBMS algorithms are tested using AVIRIS cuprite scene. To compare the performance of these algorithms, conventional PPI algorithm has also been implemented and results are obtained. The population of the dataset is based on the scene 4 of the cuprite image. The selected scene 4 for experimenting and processing is 970619t01p02\_r02\_sc04.a.rfi, and it is shown within a rectangle bound in Fig. 4.



**Fig. 4** Scene 4 of cuprite dataset

**Table 1** Skewers—SBMS algorithm

Skewer 1	Skewer 2	Skewer 3	Skewer 4	Skewer 5
0.08	0.08	0.12	0.08	0.09
0.14	0.09	0.09	0.12	0.10
0.12	0.17	0.11	0.10	0.12
0.13	0.18	0.16	0.18	0.17
0.15	0.16	0.13	0.12	0.17
0.20	0.20	0.16	0.15	0.13

**Table 2** Skewer—SBSS algorithm

Skewer 1
0.0018
0.0709
0.0883
0.0952
0.1032
0.1078

**Table 3** Skewers generated using the PPI algorithm

Iteration 1					Iteration 2				
Five skewers generated in first iteration					Five skewers generated in second iteration				
0.59	0.31	0.10	2.27	0.35	0.16	1.89	0.65	1.16	1.62
0.44	0.08	1.31	0.07	0.16	0.04	0.13	0.33	0.93	0.41
1.35	1.01	0.69	1.03	2.00	1.18	0.54	2.42	1.75	0.55
0.00	0.09	0.03	1.23	0.73	0.12	0.36	1.68	0.14	1.61
1.16	1.70	0.17	0.76	1.95	1.44	0.44	0.78	1.11	0.31
0.94	0.69	1.31	0.45	1.41	0.94	0.16	0.67	0.21	1.44

MATLAB software is used for implementing the proposed EE algorithm using HSI dataset. In the conventional PPI algorithm, the skewers were generated randomly, and therefore, the values will be changed in each of the iterations which are shown in Table 3. The skewers generated by SBMS algorithm are shown in Table 1, and similarly, the skewer generated by SBSS is shown in Table 2.

The pixel purity index calculated for the first five iterations using SBSS, SBMS and conventional PPI algorithms is given in Table 4. From Table 4, it can be observed that the pixel positions which are considered to be pure are getting varied in all the iterations for conventional PPI algorithm, whereas it remains the same for SBMS and SBSS algorithms. This improves the reliability of the result obtained and ensures the advantage of the removal of randomness by the proposed algorithm.

**Table 4** Pixel purity index generated by PPI, SBSS and SBMS

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
PPI	286914	286914	286914	286914	287929
	287929	287929	287929	287929	286914
	285891	285891	267548	267548	285891
	310135	287426	285891	285891	267454
	267548	267037	267037	310135	310135
SBSS	286914	286914	286914	286914	286914
SBMS	267037	267037	267037	267037	267037
	310135	310135	310135	310135	310135
	267548	310134	267548	267548	267548
	285891	267548	310134	285891	310134
	310134	285891	141545	287939	285891

## 6 Spectral Discrimination Measures

There are many pure pixel-based measures to evaluate the similarity between two pixels. In this section, two spectral discrimination measures that will be used to measure the similarity between source spectrum and target spectrum are presented.

### Spectral Angle Mapper (SAM)

SAM is a function which is used to find the spectral angle error between two vectors. Two parameters were given as an input to the SAM function [14] to find the spectral angle error. They are the pure pixel vectors identified by the end-member extraction algorithm and the target library spectral vector. The output of the SAM function gives the angle difference between the source spectrum and target spectrum.

$$\theta = \cos^{-1} \left[ \frac{\sum_{i=1}^n t_i r_i}{\sqrt{\sum_{i=1}^n t_i^2 \sum_{i=1}^n r_i^2}} \right] \quad (1)$$

where  $t$  is the source spectrum,  $r$  is the target spectrum,  $\theta$  is the average angle and  $i$  is the band. Angular difference is the output of SAM function which is measured in radian ranging from 0 to  $\pi/2$ . If the spectral angle value (SA) is very minute, it means that there is a high similarity between source spectra and target spectra. In the other way, if the SA value is enormous, it means that there is a less similarity

**Table 5** SAM scores of PPI, SBMS and SBSS

Minerals	SAM score		
	PPI	SBSS	SBMS
Alunite	0.287	0.285	0.283
Buddingtonite	0.106	0.101	0.085
Calcite	0.225	0.225	0.221
Kaolinite	0.228	0.223	0.218
Muscovite	0.148	0.146	0.146

between source spectra and target spectra. The SAM scores obtained for the conventional PPI algorithm, SBSS and SBMS algorithms are shown in Table 5. The source spectra values were populated using the ASTER spectral library version 2.0 library file, which was given by Jet Propulsion Laboratory (JPL), California Institute of Technology, for the purpose of research.

The smaller the SAM values across the five minerals considered, the better the results. As per the results of SAM, the pure pixel index greatly matches with the mineral buddingtonite.

## 7 Computational Complexity

PPI algorithm uses randomly generated skewers, and it generally results in different numbers of replacements for different runs. Therefore, in this paper, PPI is used to compare with SBSS and SBMS, where these two algorithms generate skewers with certain parameters discussed in Sect. 4. According to the calculation of computational complexity used in [8, 15], Table 6 lists the complexity of all the three algorithms based on floating point operation (flops), where  $p$  is the number of endmembers,  $k$  is the number of skewers,  $N$  is the number of total pixels,  $\eta$  is the number of pixels discarded after every 100 iterations. PPI algorithm requires  $10^4$  iterations, SBSS requires one iteration, and SBMS algorithm requires  $10^3$  iterations. Moreover, in SBMS, the  $N$  value will get reduced up to  $\eta$  percentage after every 100 iterations until the termination condition.

**Table 6** SAM scores of PPI, SBMS and SBSS

Algorithm	Complexity (flops)
PPI	$2pkN$
SBSS	$2pN$
SBMS	$2kp(N - \eta)$

## 8 Conclusion

This paper presents a performance evaluation of three endmember extraction algorithms, namely PPI, SBSS and SBMS. SBSS and SBMS show more relevance to the material available than PPI algorithm. As the skewers used were based on the statistical parameters of the dataset, it improves the accuracy of finding the pure pixel index. Moreover, the number of pixels has been reduced after each projection, thereby eliminating the vectors which were not considered as pure pixels. This reduced the computational complexity in a greater level and improves the running time of the algorithm.

## References

1. Goetz, F.H., Vane, G., Solomon, J.E., Rock, B.N.: Imaging spectrometry for Earth remote sensing. *Science* **228**, 1147–1153 (1985)
2. Wu, X., Huang, B., Plaza, A., Li, Y., Wu, C.: Real-time implementation of the pixel purity index algorithm for endmember identification on GPUs. *IEEE Geosci. Remote Sens. Lett.* **3**, 955–959 (2014)
3. Green, R.O., Eastwood, M.L., Sarture, C.M., Chrien, T.G., Aronsson, M., Chippendale, B.J., Faust, J.A., Pavri, B.E., Chovit, C.J., Solis, M., Olah, M.R., Williams, O.: Imaging spectroscopy and the airborne visible/ infrared imaging spectrometer (AVIRIS). *Remote Sens. Environ.* **65**, 227–248 (1998)
4. Valero, S., Salembier, P., Chanussot, J.: Hyperspectral image representation and processing with binary partition trees. *IEEE Trans. Image Process.* **22**, 1430–1443 (2013)
5. Keshava, N.: A survey of spectral unmixing algorithms. *Lincoln Lab. J.* **14**, 55–78 (2003)
6. Boardman, J.M., Kruse, F.A., Green, R.O.: Mapping target signatures via partial unmixing of AVIRIS data. In: *Proceedings of Summaries JPL Airborne Earth Science Workshop*, Pasadena, CA, vol. 1, 23–26 (1995)
7. Winter, M.E.: N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In: *Proceedings of SPIE Conference Imaging Spectrometry*, pp. 266–275 (1999)
8. Nascimento, J., Bioucas-Dias, J.: Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **43**, 898–910 (2005)
9. Liu, J.M., Zhang, J.S.: A new maximum simplex volume method based on householder transformation for endmember extraction. *IEEE Trans. Geosci. Remote Sens.* **50**, 104–118 (2012)
10. Li, J., Bioucas-Dias, J.: Minimum volume simplex analysis: a fast algorithm to unmix hyperspectral data. In: *Proceedings of IEEE Geoscience Remote Sensing Symposium (IGARSS'08)*, vol. 4, pp. 2369–2371 (2008)
11. Bioucas-Dias, J.: A variable splitting augmented Lagrangian approach to linear spectral unmixing. In: *1st IEEE WHISPERS* (2009)
12. Ifarraguerri, A., Chang, C.-I.: Multispectral and hyperspectral image analysis with convex cones. *IEEE Trans. Geosci. Remote Sens.* **37**, 756–770 (1999)
13. Graceline Jasmine, S., Pattabiraman, V.: Hyperspectral image analysis using end member extraction algorithm. *Int. J. Pure Appl. Math.* **101**, 809–829 (2015)
14. Plaza, A., Benediktsson, J.A., Boardman, J., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, J., Marconcini, M., Tilton, J.C., Trianni, G.:

- Recent advances in techniques for hyperspectral image processing. *Remote Sens. Environ.* **113**, 110–122 (2009)
15. Chang, C.-I., Wu, C.-C., Liu, W., Ouyang, Y.-C.: A new growing method for simplex-based endmember extraction algorithm. *IEEE Trans. Geosci. Remote Sens.* **44** (2006)



# A Multimedia Cloud Framework to Guarantee Quality of Experience (QoE) in Live Streaming



D. Preetha Evangeline and Anandhakumar Palanisamy

**Abstract** Cloud multimedia streaming is an evolving technology to meet intensive bandwidth that is required by conventional multimedia to stream live events. Managing device heterogeneity is critical and affects user experience drastically. Response time and bandwidth are other issues to be focused on. The streaming services involve both desktop users and mobile users. High-definition live streaming video applications are quite challenging when considering mobile devices due to their restrained handling capability and bandwidth-constrained network connectivity. To meet up with the problem of resource allocation, bandwidth allocation, fault tolerance and at the same time to guarantee the desired level of Quality of Experience (QoE) to the end users an entire framework is proposed with novel algorithms for all the above addressed issues. The resource allocation performed at the cloud-end needs to be a dynamic process. The proposed framework incorporates the novel Guess Fit algorithm to provision virtual machines dynamically based on the priority scores calculated using probabilities as a result of the combined Naive Bayes algorithm with association rule mining. The score takes into account the hit ratios and the penalty values. The proposed Guess Fit Algorithm is found to perform better than the existing Fit algorithms.

**Keywords** Resource allocation • Bandwidth sharing • Cluster computing  
Cloud multimedia • Live streaming

---

D. Preetha Evangeline (✉) · A. Palanisamy  
Department of Information Technology, SRM University, Kattankulathur, Chennai, India  
e-mail: preethaevangeline.d@ktr.srmuniv.ac.in

A. Palanisamy  
e-mail: anandh@annauniv.edu

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_15](https://doi.org/10.1007/978-981-10-7200-0_15)

## 1 Introduction

Recent technologies are being incorporated with the concept of mobility. The users view multimedia content on-demand from the servers and view them on the fly. The location based strategies and other network fluctuations should not interrupt the multimedia content being streamed. Mobile devices are usually provided with limited bandwidth. Scarcity in bandwidth needs to be handled efficiently to achieve uninterrupted services with desired level in Quality of Experience (QoE). This is one of the greatest challenges in the area of multimedia cloud. The concept of QoE is entirely exclusive to multimedia cloud environments where it differs a lot from the conventional Quality of Service (QoS). Multimedia providers can guarantee QoS levels as desired by the users but it is a truly a challenging task for them to provide the user with the multimedia services at the desired level of QoE. QoE-aware multimedia services in a cloud environment has high scope among active researchers. Traditional multimedia services include various services such as streaming, video conferencing, content sharing, real-time monitoring and broadcasting [1].

If the multimedia services rely on the client systems for their processing the results are very drastic as these services computationally require high processing capabilities. On the other hand, the development of cloud-based environment was centered to be highly distributed and to share the workload between a number of servers in the cloud [2]. Hence, there is a great hope for multimedia services to efficiently use cloud resources and deliver content to the users with the expected quality. This gave rise to multimedia cloud which entirely deals with how the user requests are processed and the needed resources are allocated. Though this may seem easy, there are additional difficulties in the multimedia cloud which do not occur in traditional cloud environments.

Peer-to-peer networks are highly used for the purpose of multimedia streaming [3]. But there is basically one problem such that networks are usually homogenous in nature. They accept a group of selected nodes or a group of mobile nodes. But the scenario considered here is heterogeneous. But it cannot be called truly homogeneous homogeneity since the selected nodes differ in their computational power, bandwidth requirements etc. And these variations in the features greatly influence the Quality of Service (QoS) parameters.

Now in such a scenario, the mobile users have to access all content from the cloud server; there is a great shortage in the bandwidth. Moreover, the multimedia service provider (MSP) has to spend more on the traffic he/she incurs. The traffic is due to the redundant requests from various mobile devices. This cost to the multimedia service provider is leveraged as high costs to the mobile content requestors. And one more parameter which affects the offered QoS to the user directly is the

distance between the user client and the multimedia service provider server. Since multimedia files are usually large in size, distance does matter and affects QoS and QoE for the video content which is delivered to the user which needs to be addressed. Live streaming application hits the criticality scenario which requires high performance and Quality of Service in order to satisfy the user experience. Mobile users try to access the content from the cloud server with minimal bandwidth that is available [4]. The cost due to redundant requests from mobile devices is leverage by the multimedia service provider as high costs to the mobile content requestors. Fault tolerance of a system is the property by which the system can recover from a failure. The fault tolerance metrics are computed using the variation in a few parameters rather than a straight forward approach in which it is necessary to compute numerous parameters. The quantification of availability in general sense. Availability is defined in terms of the service time and downtime.

$$Availability = \frac{Servicetime - Downtime}{Servicetime} \quad (1)$$

Reliability is defined as follows:

$$Reliability = \frac{No. of successful responses}{No. of responses} \quad (2)$$

Though the existing methodologies quantify mathematically and provide metrics for reliability and fault tolerance efficient approaches are still needed to ensure streaming at the required QoE level by the user.

Multimedia services require intensive computational resources. This requirement is met using a number of high-end servers with high-performance capabilities. As the number of servers increases, there is a proportional increase in the energy consumption. The scenario is a trade-off between the energy consumption and the QoE level achieved [5]. There is also an additional concern that the environment is susceptible to sudden changes in the user demand, which directly affects the QoE provisioning. Hence, any methodology developed for efficient QoE provisioning must take this point into consideration.

- Device heterogeneity (variation in resolution, processing capabilities)
- Limited bandwidth of mobile devices (~15–45 Mb/s); (>120 Mb/s in case of desktop nodes)
- Different response time requirements
- Virtual surgery needs immediate response compared to entertainment video
- Trade-off time between response time and resource cost
- Optimal provisioning of resources needed
- Failure transparency.

## 2 Related Work

### 2.1 Cloud Multimedia Streaming

Fajardo et al. [6] proposed a modular system developed in C# on the .NET framework platform. The system consists of five principal modules, namely the buffer module, the batch oracle, the transfer module, the event sender, and the event receiver. The buffer module is the point of event input/output endpoint present at sender/receiver. The batch oracle selects batch size. The transfer module performs multi-route streaming. Event sender and receiver perform serialization and deserialization. The batch size is chosen based on the latency. This approach is not dynamic and does not take into account the fault tolerance mechanisms. Lai et al. [7] proposed a network and device-aware QoS approach for cloud-based mobile streaming. When a mobile device requests a multimedia streaming service, the profile agent gathers information about its hardware and network environment parameters. Baldesi et al. [8] cloud agent determines the current status and the scope of parameters and then transmits them to the Network and Device-Aware Multi-layer Management (NDAMM).

Chen et al. [9] presented a generic framework which facilitated a cost-effective cloud service for crowdsourced live streaming. Through adaptive leasing, the cloud servers were provisioned with a fine granularity in order to accommodate geo-distributed video crowdsourcers. It also presents an optimal solution dealing with service migration among cloud instances. This approach could be further enhanced with crowdsourcer prediction through user behavior analysis from real-world measurement results. Bellavista et al. [10] proposed a dynamic and adjustable multimedia streaming service architecture over cloud computing. It adopts the device profile manager concept. This policy evaluation module determines multimedia coding parameters in terms of device profile and real-time data (Networkdata).

Sun et al. [11] proposed Cloud Media on-demand cloud resource provisioning methodology, which can meet the dynamic and intensive resource demands of VoD over the Internet. A novel queuing network model to characterize users' viewing behaviors and two optimization problems related to VM provisioning and storage rental are proposed. A dynamic cloud provisioning algorithm is designed and implemented, by which a VoD provider can effectively configure the cloud services to meet its demands. The results confirmed the adaptability and effectiveness of Cloud Media in handling time-varying demands and guaranteeing smooth playback at any time.

Chang et al. [12] proposed a novel Cloud-based P2P Live Video Streaming Platform (CloudPP) that introduces the concept of using public cloud servers, such as Amazon EC2, to construct an efficient and scalable video delivery platform with SVC technology. It addresses the problem of serving video streaming requests by using the least possible number of cloud servers. Compared to the traditional single delivery tree architecture, the proposed structure can save around 50% of the total number of cloud servers used and improve about 90% of the benefit cost rate.

## 2.2 Resource Allocation Strategies

Palmieri et al. [13] proposed GRASP-based resource re-optimization for effective big data access in federated clouds. GRASP is the proposed exploration technique. The system is divided into a set of cloud sites, data, and virtual machines (VMs). The algorithm initially aims that a greedy selection criterion must be satisfied. It maintains a Restricted Candidate List (RCL). The status 4-tuples considered and checked if re-routing through a path is successful [14]. If yes, next tuple is selected for processing, or else, migrate the VM and start the process from the beginning. This system is not very flexible with changes. Vaneet Aggarwal et al. [15] tried optimizing cloud resources for delivering IPTV services using virtualization techniques. The authors formulated the problem as an optimization problem and computed the number of servers required based on a generic cost function. This paper did have a few cons associated with it where it did not consider storage cost as a part of the cost functions. The property of generalization was not considered in the case where homogeneous servers were used.

Chang et al. [16] proposed scheduling policies suitable for swarm-based P2P live streaming systems. The first is content-diversified oriented (cd-oriented) policy, which gives equal importance to all chunks and schedules chunks in a random fashion. With this approach, peers hold different parts of stream content and contribute their available bandwidth to the system. The second is importance-first oriented (if-oriented) policy, which gives each chunk a content-dependent priority and first schedules the highest-priority chunk to be sent. In doing so, important chunks are more likely to be successfully received before their playback. This scheduling method is quite simple and requires only the integration of a simple data availability detector into each peer. Efthymiopoulou et al. [17] presented a system able to monitor and control the upload bandwidth of the participating peers in a P2P live streaming system, in a distributed and scalable way. Proposed system efficiently and stably delivers the video stream with low cost in terms of the bandwidth that it allocates. But it does not create a robust solution suitable for peer arrivals and departures.

## 3 Proposed System Architecture

Cloud-based live streaming is becoming an increasingly popular technology, with a large number of academic and commercial products being designed and deployed. In such systems, one of the main challenges is to provide a good Quality of Experience (QoE) in spite of the dynamic behavior of the network [18]. For live streaming, QoE can be derived from a number of factors and metrics [19]. In this chapter, the proposed framework to guarantee the desired level of QoE to the user is discussed. The basic elements of this architecture are streaming video server and a few other components as shown in Fig. 1. Each of the components is described below.

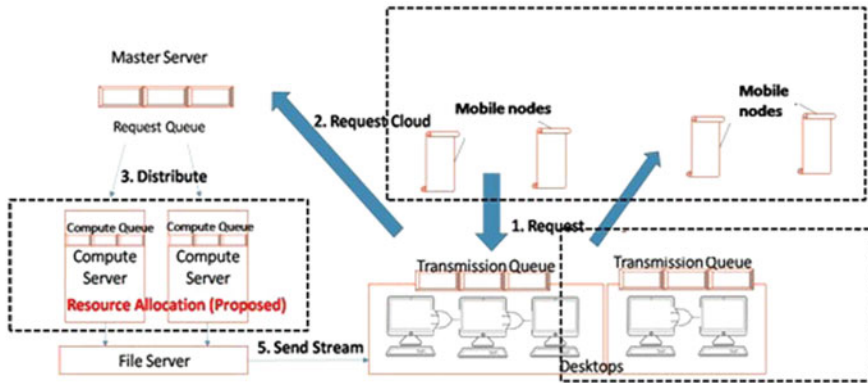


Fig. 1 System architecture

### 3.1 Queuing Module

The model consists of three concatenated queuing systems, which are the schedule queue, the computation queue, and the transmission queue. The master server maintains the schedule queue for all requests. Since two consecutive arriving requests may be sent from different users, the numbers of requests occur in non-overlapping intervals are independent random variables. The sub-flows resulting from stochastically splitting a Poisson flow are still Poisson flows. In each queuing system, the waiting time of a job in the queue corresponds to the waiting time of the request for available resources, while the service time of the job represents the real processing time of the request. Moreover, the response time, which is the sum of waiting time and service time in the queue, maps to the total time the request takes to finish the corresponding process. Therefore, the total response time for a request in multimedia cloud is the sum of the response time in the three concatenated queuing systems.

### 3.2 Resource Allocation Module

There are two types of residual capacity [20]. Reservation-based residual capacity subtracts VM resource reservations from a server's total capacity. Demand-based residual capacity is the measured amount of free resources on a server. Both can be used for placement controllers. The various approaches include First Fit, Best Fit, Worst Fit, and Next Fit. First Fit algorithms allocate the virtual machines in the order of their requests by not at all considering optimality. Even if the required resource is very less compared to the virtual machine capacity available, it gets allocated. The Best Fit strategy allocates the virtual machines from the lowest to the highest capacities. The Worst Fit assigns the highest capacity virtual machine to the

first request and continues in a similar manner. The Next Fit is similar to First Fit but does not proceed sequentially. It has a pointer parameter which points to the next virtual machine to be allocated.

### 3.3 Streaming Module

Request processing is performed by this entity, and it also helps in retrieving the video content from the File Server/Live Input and forwarding to the transmission queues found in the desktop devices. The desktop-to-mobile transfer is also performed by this module. The streaming module is adaptive and has capability to adapt itself to the content being streamed. Possible transcoding algorithms are incorporated. In case if mobile service downgrades, the streaming module has to adapt itself to continue providing services at a lower bandwidth requirement and avoid loss of transmission or breaks in the streaming process [21].

## 4 Proposed Guess Fit Algorithm

Generally, resource allocator sits at the master server in the cloud environment and allocates the necessary amount of virtual machines needed for computation and other tasks. This is accomplished using the various Fit algorithms such as Best Fit, Worst Fit, Next Fit, and First Fit. The outcome of the various above Fit algorithms has been implemented and analyzed. The Best Fit strategy allocates the virtual machines from the lowest to the highest capacities. The Worst Fit assigns the highest capacity virtual machine to the first request and continues in a similar manner. The Next Fit is similar to First Fit but does not proceed sequentially. It has a pointer parameter which points to the next virtual machine to be allocated.

Choosing the right algorithm is very critical for the performance of the streaming system.

The Guess Fit algorithm is based on predicting the occurrence of video requests based on the time slots within a day. It is based on modified Naive Bayesian classification with association rule mining. It also incorporates penalty and priority factors.

Consider  $j$  number of videos  $V_1, V_2, \dots, V_j$  and  $i$  number of time slots  $T_1, T_2, \dots, T_i$ . According to the basic axiom of probability,

$$P(A \cap B) = P(B).P(A|B) \quad (3)$$

Here, the events  $A$  and  $B$  represent:

$A \rightarrow$  Event that video of category  $V_j$  occurs

$B \rightarrow$  Event that current time belongs to Time slot  $T_i$

where  $1 \leq j \leq n$  and  $1 \leq i \leq m$

The value of  $m$  may be 2, 3, 4, 6, or any other value. These values of  $m$  correspond to 12 h, 8 h, 6 h, and 4 h periods, respectively. Based on the value of  $T_i$ , the value of probability of occurrence  $V_j$  can be predicted by the Guess Fit algorithm.

$$P(V_j \cap T_i) = P(T_i) \cdot P(V_j | T_i) \quad (4)$$

Equating,

$$P(T_i) \cdot P(V_j | T_i) = P(V_j) \cdot P(T_i | V_j) \quad (5)$$

$$P(T_i | V_j) = \frac{P(T_i) \cdot P(V_j | T_i)}{P(V_j)} \quad (6)$$

The odds form of the Bayes theorem allows to compare  $P(V_j)$  and  $P(\sim V_j)$  directly

$$P(\sim V_j | T_i) = \frac{P(\sim V_j) \cdot P(T_i | \sim V_j)}{P(\sim V_j) \cdot P(T_i | \sim V_j) + P(V_j) \cdot P(T_i | V_j)} \quad (7)$$

From dividing  $P(V_j | T_i)$  by  $P(\sim V_j | T_i)$ ,

$$\frac{P(V_j | T_i)}{P(\sim V_j | T_i)} = \frac{\frac{P(V_j) \cdot P(T_i | V_j)}{P(\sim V_j) \cdot P(T_i | \sim V_j) + P(V_j) \cdot P(T_i | V_j)}}{\frac{P(\sim V_j) \cdot P(T_i | \sim V_j)}{P(V_j) \cdot P(T_i | V_j) + P(\sim V_j) \cdot P(T_i | \sim V_j)}} \quad (8)$$

Simplifying,

$$\frac{P(V_j | T_i)}{P(\sim V_j | T_i)} = \frac{P(V_j) \cdot P(T_i | V_j)}{P(\sim V_j) \cdot P(T_i | \sim V_j)} \quad (9)$$

$$\frac{P(\sim V_j | T_i)}{P(V_j | T_i)} = \frac{P(\sim V_j)}{P(V_j)} \times \frac{P(T_i | V_j)}{P(T_i | \sim V_j)} \quad (10)$$

Assume some pre-existing training set,  $\{x^{(k)}, y^{(k)}\}$  where  $k = 1, 2, \dots, n$ .

$x^{(k)}$  is a vector and  $y^{(k)}$  has values from 1 to  $k$ .

Here,  $k$  denotes the number of classes. Obviously  $k = i$ , where  $i$  is the time slot to which the video occurrence belongs.

We have a multi-class classification (prediction) problem. Except for when  $i = 2$ , which is binary classification (prediction).



The task is to map vector  $x$  to its corresponding label  $y$ . Assume random variables  $Y$  and  $X_1$  to  $X_d$ , corresponding to label and vector components  $x_1, x_2, \dots, x_d$ .

Key idea of Naive Bayes is based on:

$$\begin{aligned}
 &P(T_i = y, X_1 = x_1, \dots, X_d = x_d) \\
 &= P(T_i = y) \times P(X_1 = x_1, \dots, X_d = x_d | T_i = y) \\
 &= P(T_i = y) \times \prod_{m=1}^d P(X_m = x_m | X_1 = x_1, \dots, X_d = x_d, T_i = y) \quad (11) \\
 &= P(T_i = y) \times \prod_{m=1}^d P(X_m = x_m | T = y)
 \end{aligned}$$

This is the Naive Bayes assumption.

The Naive Bayes algorithm is run at regular time intervals (say 24 h), collecting the necessary parameters (video, frequency, time slot) needed for future prediction. Time slots are also updated as and when necessary.

Now, to find out the possibility of video requests occurring together in a specific time slot, we apply association rule mining principles based on support and confidence measures. Every association rule has a support and confidence. Support is the percentage of requests that demonstrate the rule. An item set is called frequent if its support is equal or greater than an agreed upon minimal value—the support threshold. Here, the item set consists of videos requested. We should only consider rules derived from item sets with high support and high confidence.

The confidence is the conditional probability that, given  $X$  present in a transition,  $Y$  will also be present.

Confidence measure, by definition:

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{support}(X, Y)}{\text{support}(X)} \quad (12)$$

More clearly,

let  $S_i$  denote  $\text{Support}(V_i)$ , then

$$S_i = \text{COUNT}(V_i) \quad (13)$$

And Confidence of  $V_i, V_j$  be  $C_{ij} = S_{ij}/S_i$  where  $S_{ij}$  is the support of both  $i$  and  $j$  requests occurring.

With this derivation of various quantities needed for the Guess Fit algorithm, the algorithm proceeds as follows.

Guess Fit algorithm executes in two phases. In the first phase, initially, a First Fit approach is followed as there exists no idea about the arrival of video requests. During this phase, the algorithm collects the necessary parameter information needed. This paves way for phase two of the algorithm.

**Algorithm 1** Guess Fit

---

```

Res_req=0;
Compute P(Ti) values where  $P(T_i) = \frac{\text{No. of entries having } T_i}{\text{Total no. of entries}}$ 
Compute P(V=Vj|Ti) for all values of i and j.
Similarly compute P(Time=Timej|Ti) and P(Freq=Freqj|Ti)
Compute P(V|Ti) = P(V=Vj|Ti) x P(Time=Timej|Ti) x
P(Freq=Freqj|Ti)
Maximize P(V|Ti) x P(Ti).
for current time slot ti,
Res_req=  $\sum$  Resources needed by maximized results
Compute  $\sum C_{ji}$  based on support and confidence measures,
if(threshold>0.5) //strong association rule
Res_req= $\sum R_{ji}$  where Rji is the resource needed by video j in
timeslot i.
Res_free=Res_total-Res_req; //This is available for other
video requests
But mispredictions of videos can also be possible. For
example, due to a symposium/seminar a particular video can
be demanded for a period of time and then the request rate
drops. To account for the mispredictions, a penalty factor
is introduced. This is dynamically updated.
Let N = No. of occurrences of Vi in Naive Bayes/Association
Rule Mining Maximization results. (No. of predicted
occurrences)
N' = No. of actual occurrences of Vi
Penalty,  $P = \frac{N'}{N}$ 
where N>3 if P≤0.5, allocate (P x R) % of total resources.
This is done to avoid over allocation of resources in case
of misprediction.
if res_req ≤ res_total
allocate (P x R) % of res_total
else
//contention
P' = (1/P) + P(Vj|Ti) + Sjk
Allocate based on higher P'
end

```

---

The contention arises when the resources needed is greater than the total resources available. For this, a priority term is introduced to provide the resources first to the higher priority requests.

(1/P) is the reciprocal of penalty term. This weighs higher if the successful predictions are more for the video in the time slot. It is like a reward factor. S<sub>jk</sub> is the count (support value) of V<sub>j</sub> in all occurrences if threshold is greater than 0.5 (Strong association rules). P(V<sub>j</sub>|T<sub>i</sub>) is the probability result obtained by the Naive Bayes algorithm for the particular video.

### 5 Results and Discussion

The following results have been obtained on the cloud-end (master server), where the Guess Fit algorithm runs. Figure 2 shows the probability of video occurrences which has been predicted by means of the Guess Fit algorithm and virtual machines are reserved for videos accordingly. Time slots per day (tslot) parameter has been varied from 2 and the results are noted down. The following graph shows the

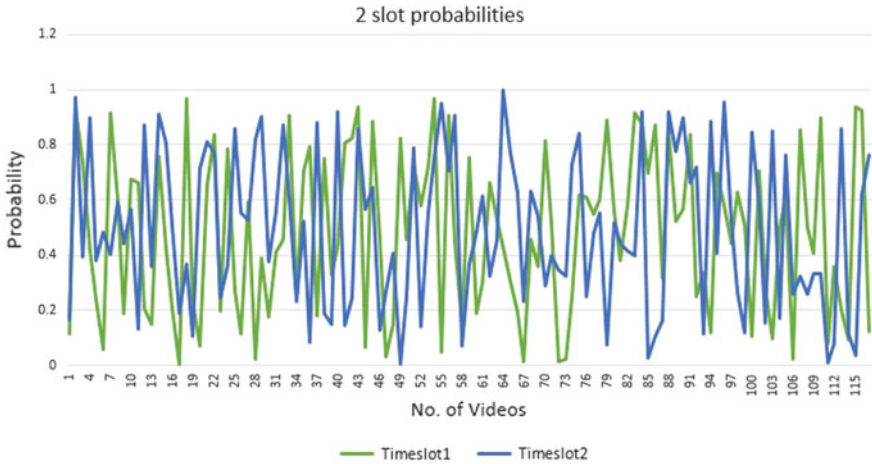


Fig. 2 Two-slot video probabilities

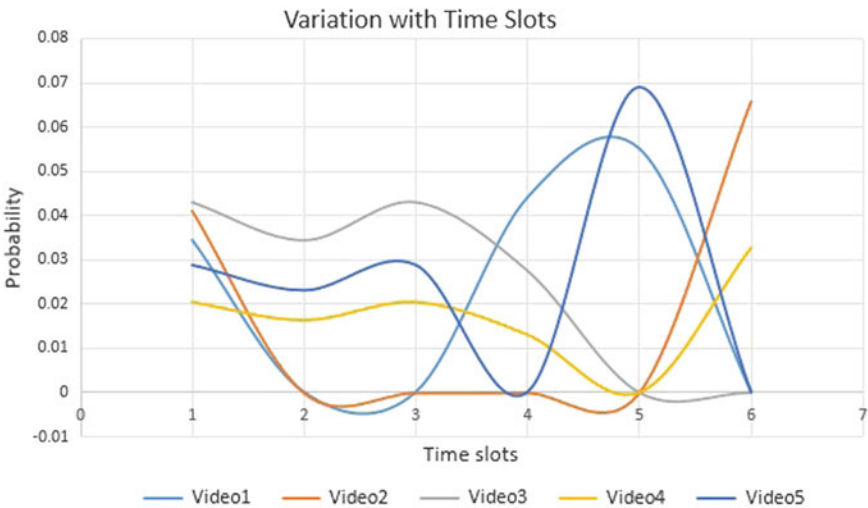


Fig. 3 Probability of videos as time slots varies

predicted probabilities for 115 videos in 2 time slots T1 and T2. Virtual machines are allocated priorly for those videos whose time slot has higher priority value.

Figure 3 shows the predicted probabilities of five different videos as the time slots are varied. For instance, video 5 has the highest probability in time slot T5 and hence VMs are allocated.

Consider the preference level of video 3. Up to time slot T4, it has the highest preference indicating that the video can be anticipated during that time duration in the day, after which the video 1 and later video 5 is preferred.

The following graph depicts the probability values of videos occurring in a time slot as the videos are varied within the time slots. The graph easily describes the expected videos and their relative preference levels within each time slot. For example, in time slot T5, video1 has the highest preference, and in time slot T3, video7 has the highest preference level in allocation of virtual machines (Fig. 4).

Table 1 shows the variation in probability of video occurrences with variation in different sets of videos. Each set of videos consisted of about 100 videos, and the

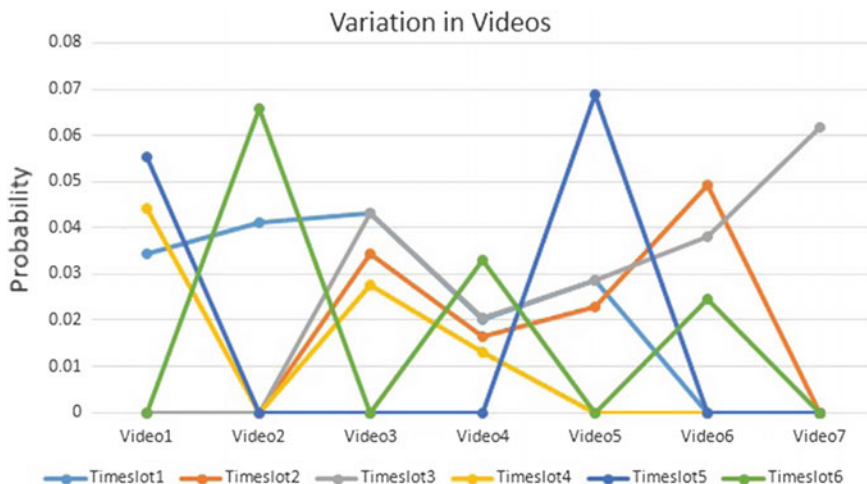


Fig. 4 Probability of videos in respective time slots as videos vary

Table 1 Probability of occurrences

Video Set	Video 1	Video 2	Video 3	Video 4	Video 5
Set 1	0.8	0.69	0.6	0.42	0.27
Set 2	0.7	0.29	0.41	0.29	0.26
Set 3	0.74	0.39	0.61	0.39	0.22
Set 4	0.61	0.43	0.51	0.39	0.57
Set 5	0.61	0.46	0.61	0.39	0.56

variation of the probability values within each set is plotted. Set3, Set4, and Set5 consisted of majority of news and current event videos and hence higher probabilities during early hours of the morning, midday, and after evening. Set1 consisted of educational-dominated videos and hence a gradual decrease until the end of the day. Set2 consisted of mixed videos (Fig. 5).

Hit ratio is the fraction of successful predictions to the total number of predictions. To avoid computation of penalty parameter and addition with the priority value, which incurs additional computational overhead, higher values of hit ratio

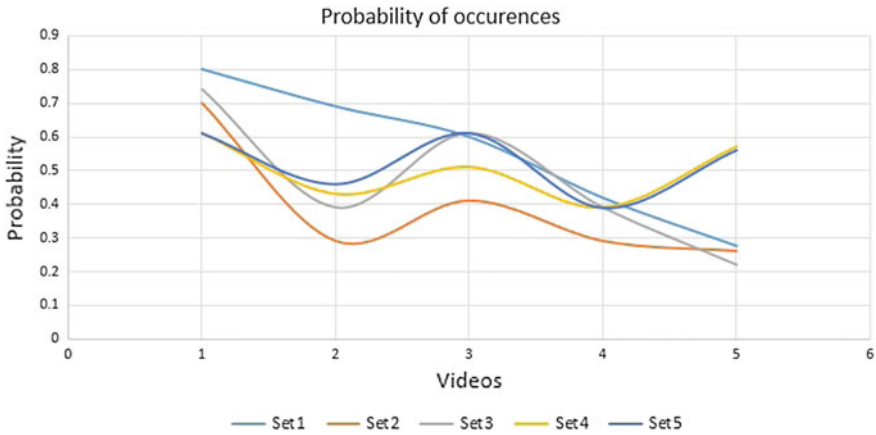


Fig. 5 Probability of occurrences

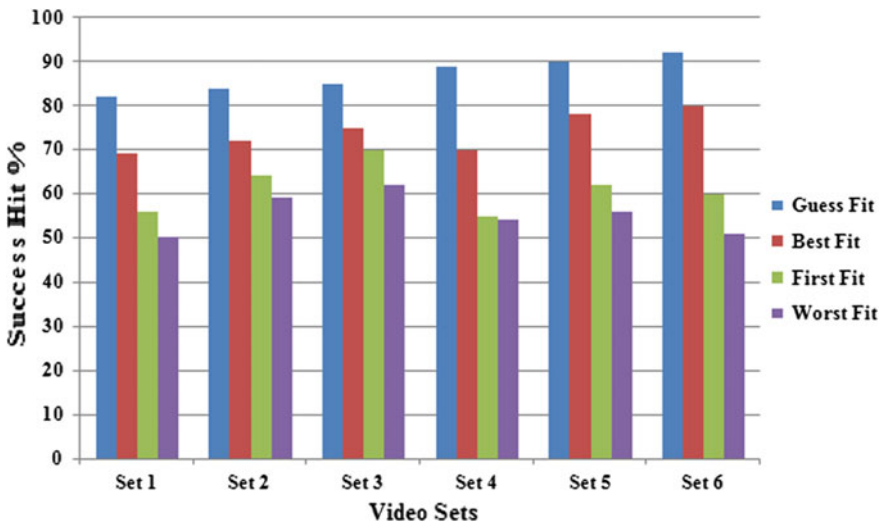


Fig. 6 Comparison of Guess Fit with other Fit algorithms

**Table 2** Variation in hit ratios as *tslot* varies

Videos	tslot = 1	tslot = 2	tslot = 3	tslot = 4	tslot = 6	tslot = 8
Set1	75	79	82	86	92	95
Set2	78	78	85	89	94	96
Set3	78	79	83	88	95	96
Set4	74	76	84	89	93	97
Set5	73	78	81	87	96	97
Set6	76	81	85	86	93	96
Set7	76	80	85	84	93	95
Set 8	72	79	85	88	94	97
Set 9	78	79	84	89	94	97

are preferred. To experiment on the outcome of hit ratios, the number of time slots per day was varied and the hit ratios were computed, and it is evident that as the number of time slots per day is increased, there are lesser mispredictions occurring and hence higher hit ratio value (Fig. 6 and Table 2).

$$\text{Hit ratio}(\text{tslot} = 8) \gg \text{Hit ratio}(\text{tslot} = 2).$$

## 6 Conclusion

The system consists of the proposed Queuing model which is built using three consecutive queues and the proposed Guess Fit resource provisioning algorithm. The prediction based Guess Fit algorithm predicts the type of video request that is expected at a particular time slot. The probability of video occurrences has been predicted by using the proposed Guess Fit algorithm and virtual machines are reserved for videos accordingly. Experiments have been carried out by varying the time slots from 2 to 8 gradually. The most relevant video for a particular time slot is guessed and resources are preferably allocated to that particular cluster playing the video. Hit ratio is the fraction of successful predictions to the total number of predictions. To avoid computation of penalty parameter and addition with the priority value, which incurs additional computational overhead, higher values of hit ratio is preferred. The probability of prediction has the Highest success rate of 96.8% when compared with other prediction algorithms. The proposed Guess Fit algorithm has been compared with other three baseline algorithms, such as First Fit, Best Fit and Worst Fit in terms of Average success Hit in predicting the video sets and allocating pre reserved resources. The proposed Guess Fit algorithm performs 35% better than Worst Fit, 12% better than First Fit and 8% better than Best Fit algorithms.

## References

1. Mehdi, S., Seyfabad, B., Akbari, D.: CAC-live: centralized assisted cloud p2p live streaming. In: Proceedings of the Twenty Second Iranian Conference on Electrical Engineering, Tehran, Iran, pp. 908–913 (2014)
2. He, J., Wen, Y., Huang, J., Wu, D.: On the cost–QoE tradeoff for cloud-based video streaming under Amazon EC2’s pricing models. *IEEE Trans. Circuits Syst. Video Technol.* **24**(4), 669–680 (2013)
3. Kliazovich, D., Bouvry, P., Khan, S.U.: Greencloud: a packet-level simulator of energy—aware cloud computing data centers. In: Proceedings of IEEE conference on global telecommunications, Austin, USA, pp. 1–5 (2010)
4. Niyato, D., Hossain, E.: Integration of WiMAX and WiFi: Optimal pricing for bandwidth sharing. *IEEE Commun. Mag.* **45**(5), 140–146 (2007)
5. Ghamkhari, M., Mohsenian-Rad, H.: Energy and performance management of green data centers: a profit maximization approach. *IEEE Trans. Smart Grid* **4**(2), 1017–1025 (2013)
6. Fajardo, J. O., Taboada, I., Liberal, F.: QoE-driven and network-aware adaptation capabilities in mobile multimedia applications. *Elsevier Mutimed. Tools Appl. J.* **70**(1), 311–332 (2014)
7. Lai, C.-F., Wang, H., Chao, H.-C., Nan, G.: A network and device aware QoS approach for cloud-based mobile streaming. *IEEE Trans. Multimed.* **15**(4), 134–143 (2013)
8. Baldesi, L., Maccari, L., Cigno, L.: Improving p2p streaming in community-lab through local strategies. In: Proceedings of the Tenth IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Lyon, France, pp. 33–39 (2014)
9. Chen, F., Zhang, C., Wang, F., Liu, J., Wang, X., Liu, Y.: Cloud-assisted live streaming for crowdsourced multimedia content. *IEEE Trans. Multimed.* **17**(9), 176–183 (2015)
10. Bellavista, P., Reale, A., Corradi, A., Koutalas, S.: Adaptive fault-tolerance for dynamic resource provisioning in distributed stream processing systems. In: Proceedings of International Conference on Extended Database Technology, Brussels, Belgium, pp. 327–334 (2014)
11. Sun, B.-J., Wu, K.-J.: Research on cloud computing application in the peer-to-peer based video-on-demand systems. In: Proceeding of IEEE 3rd International Workshop on Intelligent Systems and Applications (ISA), Wuhan, China, pp. 1–4 (2011)
12. Chang, H. Y., Shih, Y. Y., Lin, Y. W.: CloudPP: a novel cloud-based p2p live video streaming platform with svc technology. In: Proceedings of the Eight International Conference on Computing Technology and Information Management, Seoul, Korea, pp. 64–68
13. Palmieri, F., Fiore, U., Ricciardi, S., Castiglione, A.: GRASP-based resource re-optimization for effective big data access in federated clouds. *Elsevier J. Futur. Gener. Comput. Syst.* **64**(1), 168–179 (2015)
14. Wang, F., Liu, J., Chen, M., Wang, H.: Migration towards cloud-assisted live media streaming. *IEEE/ACM Trans. Netw.* **2**(99), 1–9 (2015)
15. Aggarwal, V., Gopalakrishnan, V., Jana, R., Ramakrishnan, K.K.: Optimizing cloud resources for delivering IPTV services through virtualization. *IEEE Trans. Multimedia.* **15**(4), 789–801 (2013)
16. Chang, C. Y., Chou, C. F., Chen, K. C.: Content-priority-aware chunk scheduling over swarm-based p2p live streaming system: from theoretical analysis to practical design. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **4**(1), 57–69 (2014)
17. Efthymiopoulou, M., Efthymiopoulos, N., Christakidis, A., Corazza, L., Denazis, S.: Congestion control for P2P Live Streaming. *Int. J. Peer to Peer Networks (IJP2P)* **6**(2), 1–21 (2015)
18. Goudarzi, P., Ranjbar, M.R.N.: Bandwidth allocation for video transmission with differentiated quality of experience over wireless networks. *Elsevier Comput. Electron. Eng. J.* **40**(5), 1621–1633 (2014)

19. Sardis, F., Mapp, G., Loo, J., Aiash, M.: On the investigation of cloud-based mobile media environments with service-populating and QoS-aware mechanisms. *IEEE Trans. Multimed.* **15**(4), 155–161 (2013)
20. Denning, R.: Applied R&M Manual for Defense Systems (2012). [https://www.sars.org.uk/BOK/Applied%20R&M%20Manual%20for%20Defence%20Systems%20\(GR-77\)/p0c00.pdf](https://www.sars.org.uk/BOK/Applied%20R&M%20Manual%20for%20Defence%20Systems%20(GR-77)/p0c00.pdf)
21. Fulp, E.W., Reeves, D.S.: Bandwidth provisioning and pricing for networks with multiple classes of service. *Elsevier J. Comput. Netw.* **46**(1), 41–52 (2004)



# Spectral Band Subsetting for the Accurate Mining of Two Target Classes from the Remotely Sensed Hyperspectral Big Data



H. N. Meenakshi and P. Nagabhushan

**Abstract** Due to the outstanding topical development of sensor technology, there is a substantial increase in the spatial, spectral, and temporal resolution of the remotely sensed hyperspectral data. The increase in the resolution of the data in turn has increased the volume, velocity, and variety of the data has contributed to identify the hyperspectral data as ‘Big Data.’ On the one hand, hyperspectral big data is a rich source of information for several applications as it consists of varieties of classes that include natural and man-made land covers. On the other hand, mining of the required class by the application from such a massive data turns out to be very difficult and hence requires a smart hyperspectral data analysis. Realizing that the user could be interested to mine just one or two target classes from among several classes as required by the given application, we propose an effective technique to handle this voluminous data by focusing on just one target class at a time. This research contribution includes the designing of a spectral band subsetting to address the dimensionality reduction problem by focusing on just one target class in parallel with the second target class. The proposed spectral subsetting is carried out in two stages. In the first stage, the most significant spectral band of both the target classes is instigated independently, and in the second stage, they are merged and validated. As there is a possibility that two target classes are overlapping with each other due to their spectral similarities, a method is also proposed to solve the overlapping of the target classes. The experiment is carried out on a benchmark data set, namely AVIRIS Indiana pine, ROSIS Pavia University.

**Keywords** Big data • Target class • Spectral signature • Spectral band subsetting • Overlapping classes • Density • Clusters

---

H. N. Meenakshi (✉) · P. Nagabhushan  
Department of Studies in Computer Science, University of Mysore,  
Mysore, Karnataka, India  
e-mail: meena.hn79@gmail.com

P. Nagabhushan  
e-mail: pnabhushan@hotmail.com

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_16](https://doi.org/10.1007/978-981-10-7200-0_16)

## 1 Introduction

The hyperspectral data originated from the advanced remote sensing technology like airborne and satellite has substantially increased the spectral and spatial resolution of the data (HSSR). For example, in AVIRIS from NASA/JPL, the spectral resolution of the various land cover measured is 10 nm that ranges from 0.4 to 2.55  $\mu\text{m}$  and the spatial resolution varies from 10 to 50 m acquired over several km [1]. Such hyperspectral data is constantly being sensed over a time which is required for a monitoring purpose. Further when it is let to accumulate for many hours to several days', then it results in a voluminous data (terabytes-Zetabytes). The spatial, spectral and temporal resolution of such remotely sensed data that consists of varieties of land cover classes (ranges from 5 to 50 classes) contribute to the term called 'Big data' [2].

In recent days, the increase in the importance of the hyperspectral data is due to the presence of several valuable land cover classes distributed at different locations that are necessary for various applications. Classification of the varieties of land cover classes is a frequently carried out task in hyperspectral data. However, classifying all the classes from such a voluminous data turns out to be very difficult. Hence, such massive data if not analyzed and processed smartly, then, the treasure of information would become inaccessible to the user. Moreover, an application would not focus on all the classes for its purpose instead one or at most two classes called target class. Based on the customer interest, a smart analysis of the big data has become a trend in recent days. Realizing the need by the application, mining/classifying just one target class at a time from the massive data could be very efficient. In most of the land monitoring applications, the need may be to classify at most two target classes. Mapping the tree and grass-pasture class, water bodies and marshy land, wheat and grass-pasture, railway track and roadways in a hyperspectral image that consists of quite a large number of classes are some of the examples that demand the classification of two target classes. However, one perceptible challenge while mapping the two target classes is the spectral signature similarities of the classes to be mapped which results in overlapping of the classes. Particularly, if the data is taken to represent a scene from an agricultural land cover then, classes like wheat, corn, grass-pasture, hay, and trees overlap with each other. The complexity further increases, due to the temporal nature of the data coupled with high dimensionality. The large spectral bands of the class not only increase the computational complexities but also fail to categorize them in turn affect the classification results of the intended classes. Therefore, selection of the appropriate spectral signatures is necessary not just for distinguishing the two target classes but also for an effective discrimination of the intended classes from all other classes that are present in the entire population. Naturally in such circumstances, except the knowledge of the target class, other details like overlapping classes, number of classes present in the entire input, existence of the user required target class in the big data itself and their training may not be available.

Therefore, in this paper, utilizing the knowledge of both the target classes, the optimal spectral band selection for two target class is proposed in two stages. The first stage is the prediction of the significant spectral band of the individual target

class that are carried out in parallel, and the second stage is the model designed to correct the predicted spectral bands to handle the overlapping classes. Rest of the paper is organized as follows: Sect. 2 describes the procedure for optimal spectral band selection and Sect. 3 outlines the experiment and results followed by conclusion.

## 2 Proposed Two Target Classes Driven Spectral Subsetting

Given  $\{T_1\}_{P \times N}$  and  $\{T_2\}_{Q \times N}$ , the instances of two target classes where,  $N$  is the size of the spectral bands ( $>100$ ) that defines the spectral signature of the class with a high spectral resolution. Suppose, both the target classes to be mapped are spectrally divergent then, the problem of dimensionality reduction could be solved in a single stage by determining the most desired subset of the spectral band required for its accurate projection that maximizes the cohesiveness among its samples [3]. However, if the two target classes to be mapped encompass a similar spectral signature, then they get overlapped with each other and would lead to misclassification. Hence, the problem of spectral subsetting can be formally stated as the process of determining the most desired spectral signature that increases the homogeneity within the target class and also maximizes the proximity between the two target classes. Further, the intrinsic subspace obtained from both the target classes when used to project the entire hyperspectral input, the proximity between the target classes and all other classes present in it is also expected to maximize so that, when it is subjected for classification, the false acceptance and the false rejection in both the target classes are avoided. Formally, the problem is to find the best possible combination of spectral bands such that i. the compactness within each target class should be maximized and also ii. the distance between both the classes is expected to be maximum. The compactness denoted as  $\Delta$  is a measure of the ratio between the average distance of all the samples in a class to its mean and diameter of that class as in (1). Similarly,  $\delta$  is the measure of distance between the mean of both the classes as in (2).

$$\Delta_T = \frac{\frac{1}{P} \left( \sum_{i=1}^P (x_i - \mu_T) \right)}{\max d(x_i, x_j)} \quad (1)$$

and

$$\delta = d(\mu_{T_1} - \mu_{T_2}) \quad (2)$$

Figure 1 shows the architecture for mining the target classes from the big data. The control set of the target class to be mapped undergoes the dimensionality

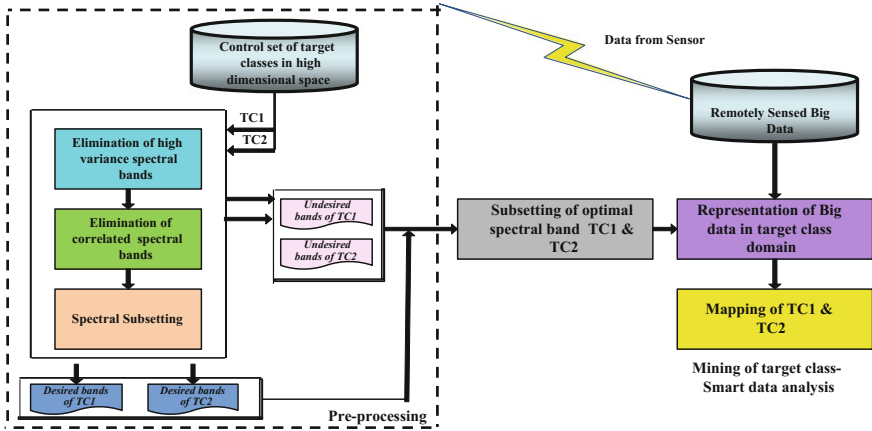


Fig. 1 Architecture for mining the target classes from remotely sensed big data

reduction in parallel, where individually it is searched for the desired and undesired spectral bands for each target class. Next, it is fine-tuned to generate the optimal subset to represent both the target classes. Upon the availability of the input data, it is projected on to the target class-derived spectral subspace and then classified.

### 2.1 Spectral Subsetting

Statistically,  $\Delta_T \propto \min(\text{var})$ , whereas  $\delta \propto \max(\text{var})$ , and both these conditions are contradiction to each other. Exploring all the possible spectral combinations in a hyperspectral data that could satisfy both the criteria leads to NP hard problem. Therefore, a simple but an efficient approach is a two-stage model. In the first stage, for each target class, the optimal spectral bands are determined based on the low-variance bands, and the steps are summarized as:

- Step1: *for* i=1:2 //for two target class
- Step2: Find the variance of the  $N$  spectral bands for an  $i^{\text{th}}$  target class
- Step3: Sort the spectral bands in the increasing order of the variance
- Step4: Eliminate all the bands whose variance is greater than the threshold and eliminate the correlated spectral bands  
that results in  $k$  spectral bands.
- Step5: Initialize  $\{d_i\} = \emptyset$  and  $\{u_i\} = \emptyset$  // initialize desire and undesired subset
- for* j=1: k
- Step6: Compute  $\Delta_j$  after projecting all the target class samples in  $j^{\text{th}}$  spectral band
- Step7: If  $\Delta_j$  is maximized, then include it in  $\{d_i\}$  else include it in  $\{u_i\}$
- end*
- end*

The method for estimating the threshold on the variance can be referred from [4]. Each target class when projected on the desired spectral subset the compactness is expected to be maximized. Let,  $|d_1| = n'$  and  $|d_2| = n''$  be the desired spectral bands of individual target classes resulted from the first stage of learning phase, then further, joining these desired spectral subset as  $D = \{d_1\} \oplus \{d_2\}$  could be the optimal subset where,  $|D| = n|n < N$ .

## 2.2 Spectral Band Subset Enhancement

Suppose,  $\{T_1\}$  and  $\{T_2\}$  are the overlapping classes then, the desired spectral subset obtained from the previous step cannot discriminate them. Since, the common spectral range that exists between these classes is the cause of overlapping of the classes where the desired subset- $\{D\}$  becomes inadequate in discriminating them. Hence in the second stage, the common spectrals are determined from the desired subset which is further corrected by utilizing the undesired subset. The procedure for correcting(enhancing) the spectral subset is as described below.

```

Step1:   find  $E = \{d_1\} \cap \{d_2\}$ 
Step2:   if  $E = \emptyset$  then consider  $\{D\}$  as the optimal subset and stop
Step3:   else if  $E \neq \emptyset$  then
Step4:   for  $i=1$  to  $|E|$ 
Step5:   find the range  $[\underline{L}_i, \bar{L}_i]$  for both  $\{T_1\}$  and  $\{T_2\}$ 
Step6:   if the spectral range overlaps then eliminate the  $I_i$  band from  $\{D\}$ 
Step7:   for  $j=|u_1|$  to 1
Step8:   for  $k=|u_2|$  to 1
Step9:   Project  $\{T_1\}$  and  $\{T_2\}$  onto a intrinsic feature subspace  $\{D\} \cup \{I_j\} \cup \{I_k\}$  and
measure  $\delta$ 
if  $\delta$  is maximized stop
else repeat step7 to step9
end
end
Step10: else consider  $\{D\}$  as the optimal subset and stop
end
end

```

## 2.3 Validation of Spectral Band Subset for Optimality

Although  $\Delta$  and  $\delta$  are used as a validation measure during the learning phase, the post-learning validation can be employed to test for the classification results assuming that the input population is not known. Since the number of the target

class is a priori known, the spectral bands can be validated by applying the K-means clustering [5] in  $\{D\}$  intrinsic feature space. For  $k=2$ , it is expected that the k-means results in two clusters such that both the target classes are well separable.

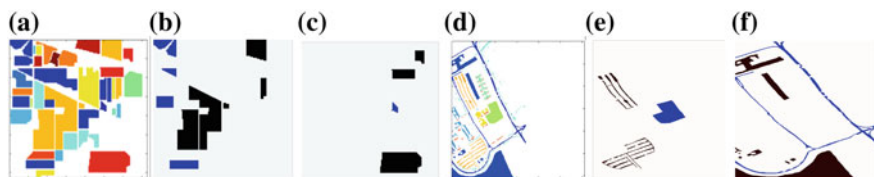
### 3 Experiments

An experiment was carried out on the AVIRIS Indiana Pines and ROSIS Pavia University hyperspectral data. The original Indiana pine data set is very large consisting of  $1475 \times 1432$  pixels, where each pixel is 20 m spatial resolution, the spectral resolution is 220, and the total number of classes is 26. Since it was taken in 1992, temporal series is not available. There exist several flight reflectances of the land covers that are temporal in nature but since our objective is not to analyze the temporal changes of the land cover, we have considered the standard benchmark data set to explore the feasibility of the proposed dimensionality reduction on big data. The detail is given in Table 1, and further details can be referred from [5].

Both the data sets are good examples of big data as they have high spectral and spatial resolution but temporal series is not collected. Pavia University does not have much overlapping classes, whereas Indiana pine is a scene acquired to cover the vegetation has many overlapping classes and hence is a challenge for classification. From each data set, only few selected classes were chosen in pair as target classes for the experimentation purpose based on the spectral similarities that are located at different places as shown in Fig. 2.

**Table 1** Data set description used in the experimentation

Data set	Spectral bands	Spectral bands after calibration	Spatial description	Classes
ROSIS Pavia University	220	200	$610 \times 610$	16
AVIRIS Indiana pine	103	100	$145 \times 145$	9



**Fig. 2** Two target classes chosen from AVIRIS Indiana Pine data set: **a** ground truth-16 classes. **b** Corn-mintill and soyabean-mintill **c** alfalfa and woods **d** ground truth of ROSIS Pavia University —9 classes **e** self-blocking bricks and bare soil **f** asphalt and meadows

The experimental set up is as given below:

---

```

DATA_SET = { Indiana Pine, Pavia_university;} // Two Hyperspectral data set
Classifier = {SVM, K-means} // Support Vector Machine to classify
after feature subsetting
1 #pragma omp parallel for // parallel model: creation of two threads to handle TC1
  & TC2
2 for i=1 to 2 do // for two data set
3   D=DATA_SET;
   Normalize(D);
4   Choose any two Target classes as TC1 & TC2; // total number of classes in the data set
5   Find the desired and undesired spectral bands for TC1 and TC2
6 end
7   Check for the overlapping of the classes and their spectral range
8   {OFS} = Find the optimal spectral subset that can separate TC1 And TC2
9   Classification Results(TC1,TC2)=SVM_Classify(D);
10 end

```

---

Data were used for experimentation after normalization. From each data set, different classes were chosen as the target classes in combination.

**Parallel Model:** To search for the desired spectral band for each target class, *pragma* directives from the *openmp* parallel programming was used. It was employed to create the two threads where each thread searches for the desired and the undesired subset of both the target classes in parallel. Three tasks were carried out in parallel as shown in Fig. 1. The experimentation was carried out for each pair of target class by varying the training set from 10% of total sample space to –100% of samples space. In the first stage, all the spectral bands that have large variance and tend to increase the scatterness within the target class are searched. The variance was calculated, and all the spectral bands whose variance is greater than the threshold was eliminated. Threshold on variance was chosen to be  $0.1 + \min(\text{variance})$ . All the correlated bands were also eliminated. From the remaining bands, the desired and undesired spectral subset was determined for a satisfactory value of  $\Delta$ . The number of bands preferred for elimination based on variance and correlation is as shown in Table 2.

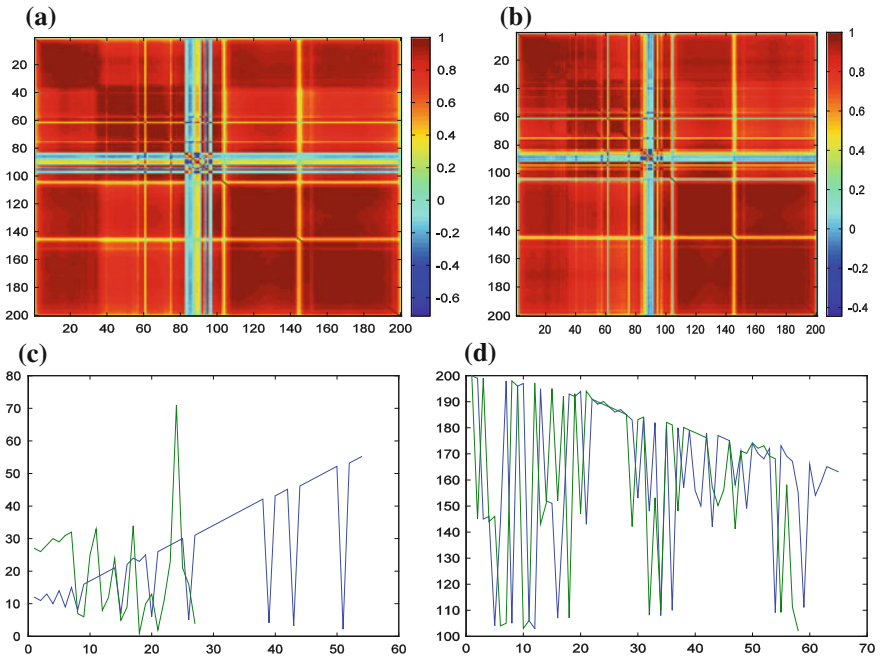
Figure 2a and b highlights the correlation among the spectral bands of corn-notill and mintill classes. It is a colormap of the individual spectral bands. Further, the overlapping of the desired spectral band subset is as shown in Fig. 2c and d.

About 20% of the desired subset was overlapping and also the value of  $\delta$  was not sufficient to separate them. Thus, the undesired subset was utilized to improve the subset and further  $\delta$  was measured.

**Optimal Spectral Subsetting:** The desired spectral bands as decided by the target classes were joined, and then, K-means clustering was applied on the control set of the chosen target classes. On the clustered data,  $\delta$  and false acceptance of one target class into the other target class was measured to check the overlapping. Next, the undesired spectral bands were iteratively explored to fine-tune the spectral subset for its optimality. Figure 3 depicts the overlapping of the corn-notill and soya-mintill target classes. Figure 4 is plot of spectral selection versus  $\delta$ —the

**Table 2** Results of spectral elimination and the consequent spectral subset

Data set	Target class	Spectral bands with max(var)	Correlated spectral bands	$\{d\}$	$\{u\}$	$\{E\}$	Overlapping $[L_i, \bar{I}_i]$
Indiana pine	Alfalfa-woods	68 56	45 36	45 51	42 57	12	8
	Corn-min Soya-min	33 41	30 29	65 55	72 75	69	60
Pavia University	Self-block	23	34	22	21	4	–
	Bare soil	29	20	32	19		
	Asphalt meadows	21 26	22 19	36 40	21 15	3	–

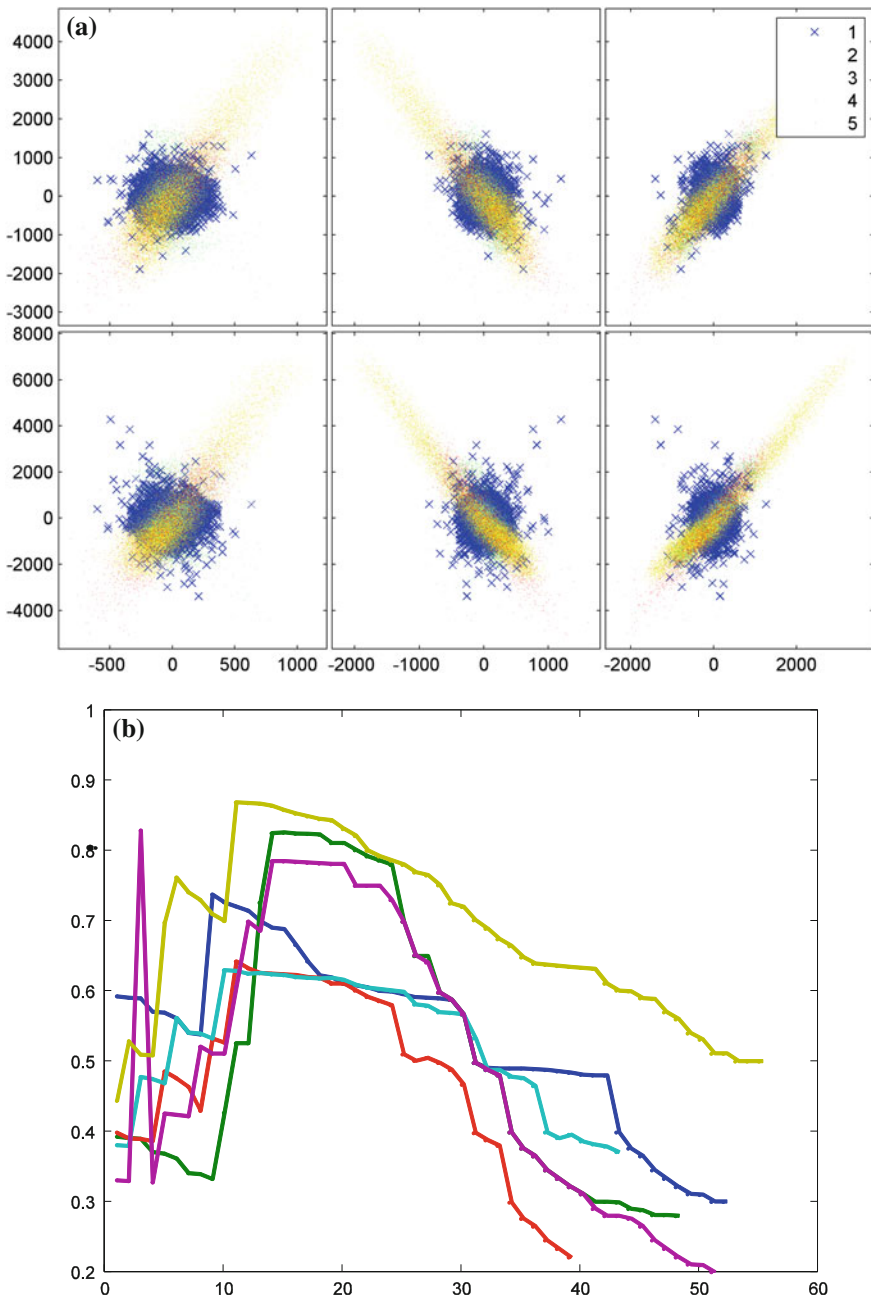


**Fig. 3** a, b Correlated spectral bands of corn-mint and soya-mintill c, d Overlapping of the spectral band of alfalfa-woods and corn-mintill–soya-mintill

distance between the centroids of the clusters. It shows the cutoff point to decide the optimal spectral bands by varying the undesired spectral. Some of the boundary samples were overlapping demanding for further correction of the subset. But in this work, spectral correction does not focus on boundary samples which could be taken as a future work.

**Mining for the Target Class:** First, the high-dimensional input is normalized and projected on to the feature space as computed from the above procedure. Support





**Fig. 4** **a** Overlapping of corn-mintill-soya-mintill **b** Undesired bands versus distance between the target classes

**Table 3** Classification results of the target class from the input population when projected on the target class-derived subspace

Data set	Target class 1	$\zeta$	$\Upsilon$	$\Xi$	Target class 2	$\zeta$	$\Upsilon$	$\Xi$
Indiana pine	Alfalfa	0.897	0.880	0.881	Woods	0.898	0.880	0.802
	Corn-min	0.81	0.8	0.710	Soyabean-min	0.78	0.85	0.79
Pavia	Self-blocking	0.998	0.913	0.956	Bare soil	0.999	0.946	0.906
University	asphalt	0.999	0.884	0.918	Meadows	0.999	0.902	0.857

machine vector is applied to classify the entire input, and results are measured for the target classes. Feature subspace was also tested for classification results using Eq. (3). Total  $145 \times 145$  input samples were first represented in the feature subspace derived by the target class based on the boundary estimated for both the classes the classification was performed to categorize both the target classes. The results of alfalfa and woods class are as shown in Table 3. The other samples were discarded. A similar experiment was carried out on other chosen pair of classes from both the data sets. It was observed that Corn-mintill and Soyabean-mintill are chosen as target classes; almost 80% of the desired spectral bands from both the classes were common out of which 70% were in the overlapping range leading to misclassification.

$$\begin{aligned}
 \text{Sensitivity} &= \zeta = TP / (TP + FN) \\
 \text{Specificity} &= \Upsilon = TN / (TN + FP) \\
 \text{Precision} &= \Xi = TP / (TP + FP)
 \end{aligned} \tag{3}$$

## 4 Results and Analysis

To analyze the effectiveness of the proposed two target class driven dimensionality reduction, a supervised contemporary dimensionality reduction was used. About 40% of the training set from all the classes were selected to find the optimal features using wrapper-based Sequential Forward Feature selection method. SVM [6] and K-means (K = 16 for Indiana pine, K = 9 for ROSIS) [4] classifier were wrapped with the feature reduction, while choosing the features, target class and classification results were measured for the entire classes Table 4.

Although the dimensionality reduction was supervised, the false acceptance and the false rejection into the required target class were more. For example, for a pair of alfalfa and woods, the results obtained from the proposed method were better when compared to the supervised SVM and K-means as it could be compared from Tables 3 and 4. Although in the proposed technique, the spectral subset was refined during the second stage, the optimal subset could not accomplish 100% for any pair of classes from both the data sets, and several false acceptance and false rejection

**Table 4** Classification results of the classes when the input was dimensionally reduced in supervised learning and classified using support vector machine and K-means classifier

Data set	CoI	K-means classifier			SVM		
		$\zeta$	$\Upsilon$	$\Xi$	$\zeta$	$\Upsilon$	$\Xi$
AVIRIS Indiana pine	Alfalfa	0.517	0.615	0.651	0.761	0.995	0.263
	Corn-notill	0.600	0.794	0.506	0.801	0.994	0.920
	Corn-mintill	0.626	0.9933	0.81	0.728	0.994	0.855
	Corn	0.513	0.99	0.469	0.898	0.990	0.509
	Grass-pasture	0.623	0.994	0.743	0.776	0.992	0.698
	Grass-trees	0.765	0.995	0.865	0.928	0.994	0.849
	Grass-pasture-mowed	0.535	0.998	0.267	0.75	0.997	0.256
	Hay-windrowed	0.744	0.996	0.841	0.744	0.996	0.831
	Soybean-notill	0.624	0.984	0.726	0.873	0.984	0.737
	Soybean-mintill	0.529	0.978	0.763	0.773	0.979	0.833
	Soybean-clean	0.554	0.995	0.768	0.824	0.995	0.846
ROSIS Pavia University	Asphalt	0.891	0.912	0.898	0.901	0.811	0.810
	Meadows	0.899	0.902	0.912	0.809	0.892	0.906
	Gravel	0.901	0.890	0.991	0.923	0.809	0.980
	Trees	0.900	0.897	0.913	0.931	0.905	0.899
	Painted metal sheets	0.890	0.899	0.901	0.912	0.902	0.911

were found in both the classes. In Pavia University data set, however, there are no overlapping classes, and hence, good results were reported.

It can also be observed that with SVM, the time required to search for the optimal subset exponentially increases owing to the increase in varieties of classes and the same was also observed with K-means classifier while searching for the optimal subset.

## 5 Conclusion

An approach for spectral band selection is developed and demonstrated to address the big data problem when just two target classes require to be mapped using their training set. In the first stage, initial optimal spectral bands are obtained and validated. If the target classes are overlapping, then the spectral selection is refined in the next stage so as to minimize the classification error. The experimental results of AVIRIS Indiana pine data set shows that false acceptance and false rejection in both the target classes got minimized due to the spectral correction process incorporated in the second stage when the overlapping classes were chosen in combination. Yet, the proposed work has not considered the issues like spectral transformation and spectral clustering focusing two targets classes and requires exploring. The big data represented in the intrinsic feature space as decided by the two target classes based

on the proposed method can be further modeled to handle the large spatial problem. The proposed work is intended for any big data but requires a testing on other big data domains like online retail and social networks.

## References

1. Pal, M., Foody, G.M.: Feature selection for classification of hyperspectral data by SVM. *IEEE Trans. Geosci. Remote Sens.* **48**(5), 2297–2307 (2010)
2. Cavallaro, G., Riedel, M., Richerzhagen, M., Benediktsson, A.J., Plaza, A.: On understanding big data impacts in remotely sensed image classification using support vector machine methods. *IEEE J. Selected Topics Appl. Earth Obs. Remote Sensing* **8**(10), 4634–4646 (2015)
3. Nagabhushan, P., Meenakshi, H.N.: Target class supervised feature subsetting. *Int. J. Comput. Appl.* **91**, 0975–8887 (2014)
4. Mac Queen, J.: Some methods for classification and analysis of multivariate observations. *Le Cam*, pp. 281–297 (1967)
5. AVIRIS Images, Jet Propulsion Laboratory, NASA. [http://aviris.jpl.nasa.gov/html/aviris\\_overview.html](http://aviris.jpl.nasa.gov/html/aviris_overview.html)
6. Qi, Z., Tian, Y., Shi, Y.: Robust twin support vector machine for pattern classification. *Pattern Recogn.* **1**(46), 305–316 (2013)

# Semantic-Based Sensitive Topic Dissemination Control Mechanism for Safe Social Networking



Bhuvaneshwari Anbalagan and C. Valliyammai

**Abstract** Online Social Networks (OSN) contains a huge volume of publicly available information shared by the users. The users tend to share certain sensitive information which can be easily leaked and disclosed to unprivileged users. It clearly clarifies that the user lacks the knowledge of access control mechanisms available to prevent information leakage and data privacy. There is a need to automatically detect and protect the information disclosed beyond the existing privacy settings offered by OSN service providers. An automatic Semantic-based Sensitive Topic (SST) sanitization mechanism is introduced in this paper, which consider user's relationship strength and semantic access rules concerning the sensitivity of the information shared on Twitter. The interaction documents undergo Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (SST-LDA) clustering to identify sensitive topic clusters. The experimental result shows (i) the topic clusters are discovered by means of cluster entropy with very high accuracy, (ii) the probability distribution of Kullback–Leibler (KL) divergence between sensitive and sanitized Twitter post leads to a very negligible information loss up to 0.24 which is practically acceptable, and (iii) the sanitization for 16 sensitive topics between 790 Twitter users is tested which can be correlated with the advanced privacy settings to the OSN users in near future.

**Keywords** Social networking · Big data · Topic modeling · Sensitivity analysis · Entropy · KL divergence

---

B. Anbalagan (✉) · C. Valliyammai  
Department of Computer Technology, MIT, Anna University, Chennai, India  
e-mail: bhuvana.cse14@gmail.com

C. Valliyammai  
e-mail: cva@annauniv.edu

## 1 Introduction

Social network refers to a social structure that consists of individuals called nodes. These nodes are tied together by certain types of dependency such as friendship, common interest, dislike, knowledge. Social network analysis considers social relationship and formulates it as a network consisting of nodes and ties [1]. The huge fragmentation of the social graph is the major issue to the usage of social network data into various proprietaries. The security has many blunders that can involve leaking the information to unknown users. At times, a person might want to hide certain information from a particular friend or from groups of friends via privacy settings [2]. In order to mitigate the difficulties caused by manually setting up access rights for each user type concerning each resource, the methodology should automatically analysis the sensitivity of the information shared and also the relationship strength between the users by analyzing the semantics, i.e., the meaning of the sensitive information being shared [3]. In this paper, privacy provisioning is carried out by analyzing sensitivity and semantics of the topics. The rest of the paper is organized as follows. Section 2 covers the related work. The proposed dissemination control model is elaborated in Sect. 3. Our experimental settings and result analysis are discussed in Sect. 4. Finally, the paper concludes with future work in Sect. 5.

## 2 Related Work

Recently, OSN attracted much attention of research community in providing a high level of privacy to the vulnerable information shared in order to protect the information from being leaked, e.g., [4]. Our proposed work is quite different from various pioneering research works on famous OSNs Twitter and Facebook because we try to relate the privacy-based differences of conventional blogs and modern sensitive social media posts by user. In terms of topic modeling, our model is based on [5] but process the entire document to determine the sensitivity of the topic and its significance semantically. Our proposed model differs from the models studied in [6], and we incorporated the access rule generator which improves the performance. In addition, no prior work has related topics based on semantic, sensitivity, and cluster entropy. The nature of OSN unstructured data makes our work harder than previous studies [7, 8] because the sensitive information must be identified to sanitize. The data set is collected as chunked short messages which are different from traditional documents in terms of volume, velocity, and veracity. A piece of work [9] tries to explore the necessity of privacy on the popular microblogging sites, which also has a different motivation than our work.

### 3 Proposed Methodology

#### 3.1 SST Dissemination Control Model

The interaction documents between the OSN users are considered, and a relatedness measurement is carried out for similar documents concerned with a particularly sensitive topic. The proposed model binds the relationship strength calculation and the semantic annotation using Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) which is shown in Fig. 1. The profile similarity and document similarity are used to estimate the overall relationship strength among the users relating to particular sensitive topics. During LSA semantic annotation, the cosine similarity of terms and relative term frequency (TF-IDF) is calculated. The entire document is reduced through Single Value Decomposition (SVD) matrix. The sensitive terms are annotated with similar corpus representing generic terms by performing the semantic mapping. The annotated message is then stored in a database for future reference. The semantic disambiguation of the noun phrases is carried out to identify the noun that comes under aggregated (semantically) similar noun. The access rules for every user are specified by the user during OSN privacy settings. The sensitive topic  $st_i$ , the User Categories  $uc_i$ , the access levels  $al_i$ , and the relation strength  $rs_i$  of the particular sensitive topic build the Access Rules  $rule_i = \langle st_i, uc_i, al_i, rs_i \rangle$  of users. The access rule forms the basis for checking the automatic user privacy setting. The documents are clustered using SST-LDA to group all the related documents.

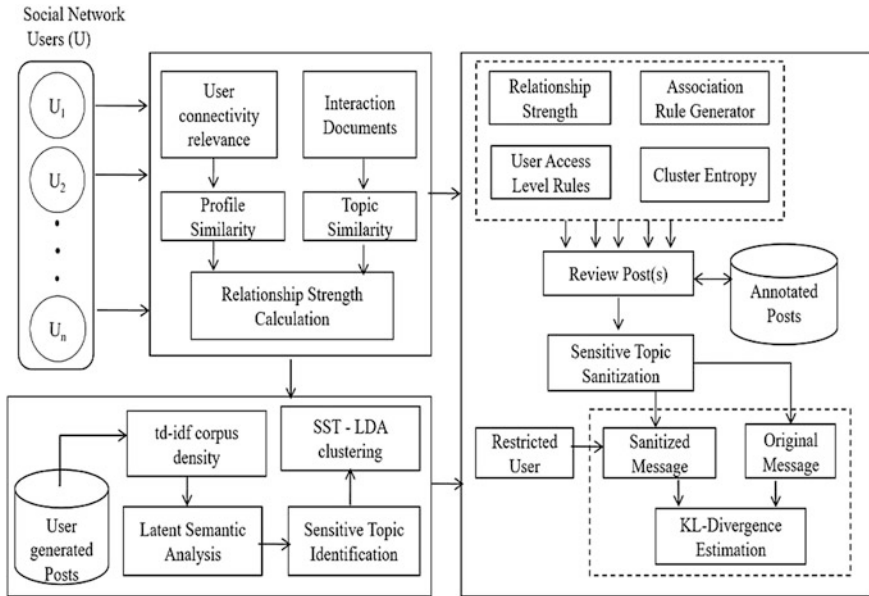
The Normalized Google Distance NGD ( $p, q$ ) is used to calculate the similarity between two search terms 'p' and 'q' which can be given as

$$NGD(p, q) = \frac{\max \log f(p), \log f(p) - \log f(p, q)}{\log M - \min \log f(p), \log f(p)} \quad (1)$$

where  $M$  is the total number of pages searched by Google,  $f(p)$  and  $f(q)$  are the number of hits for search terms 'p' and 'q,' and  $f(p, q)$  is the number of Web pages on which both  $p$  and  $q$  occur. The cosine similarity measure  $Sim(doc, c)$  between the document and the cluster is given by the similarity distance measure that can be given as

$$Sim(doc, c) = \frac{WF^m \cdot WF^n}{\|WF^m\| \cdot \|WF^n\|} \quad (2)$$

where  $WF$  is the word frequency for  $m$ th cluster and  $n$ th document. The cluster (c) identified through cosine similarity is highly related to a sensitive topic (t).



**Fig. 1** Semantic-based sensitive topic dissemination control model

The cluster relatedness  $R(c, t)$  is estimated as the product of word frequency  $Freq(w)$  and Normalized Google Distance  $NGD(w, t)$ .

$$R(c, t) = \sum Freq(w) * NGD(w, t) \tag{3}$$

The strength  $S(topic)$  of the interaction document with respect to certain sensitive topic between two users  $U_i$  and  $U_j$  is given as

$$S(topic) = \sum relatedness(doc, t) * u_{(d,i)} * u_{(d,j)} \tag{4}$$

The overall relationship strength  $RS(U_i, U_j)$  between two users with respect to a particular sensitive topic from Eqs. (3) and (4) is given as

$$RS(U_i, U_j) = Sim(i, j) + S(topic) \tag{5}$$

When the relationship strength exceeds a mean threshold level for the particular sensitive topic, the access rules are compared with an user category and the original message is sanitized using Algorithm (1).



**Algorithm (1):** Accessing Message Using Semantic Annotation

```

Input : Request to access annotated message | Output : Message M
Get Annotated message  $M_A$ , Reader classification  $R_{classify}$ , Privacy  $Priv_{req}$ 
Identify Topic A for  $M_A$ , Get the Relationship Strength (Eq.5) for A
if( $RS_{i,j} >$  threshold t)
     $\forall$ sensitive topics  $st_i$  (Eq.4) in  $ST = \{st_1, st_2, \dots, st_n\}$ ,  $uc_i$  in User Category
     $UC = \{uc_1, uc_2, \dots, uc_n\}$  and  $al_i$  in Access Level  $AL = \{al_1, al_2, \dots, al_n\}$  do
        define Access Rules  $rule_i = \langle st_i, uc_i, al_i \rangle$ 
    Identify the user type  $U_i$ , Identify the rule of  $U_i$  as  $rule_{ut}$ , Check AL in  $rule_{ut}$ 
    if( $AL$  in  $rule_{ut}$ ) return M
        else  $\forall$  sensitive terms, Replace with generic terms.
    end if. Store the modified message M modified as M
end if return M
    
```

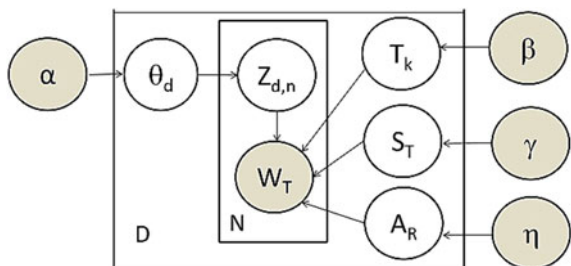
**3.2 Topic Discovery and Clustering**

The topic discovery consists of  $k$  topics  $T_k$ , for each represented by a word distribution for the topic as  $W_T$ . Let  $\theta_d$  denote the topic proportions per document and  $Z_{d,n}$  represent per word topic assignment. The sensitive topic  $S_T$ , Access Rule  $A_R$  is combined along with  $W_T$  to discover precise topics based on the word distribution with  $\alpha, \beta, \gamma, \eta$  are the Dirichlet parameters. The proposed SST-LDA model is shown in Fig. 2.

Let the total number of words ‘N’ in document ‘D’, the word distribution on topic  $W_T$  is estimated using TF-IDF. The inverse document frequency  $idf(t)$  is defined as

$$idf(t) = \log \frac{|D|}{1 + |\{d: t \in d\}|} \tag{6}$$

**Fig. 2** Proposed SST-LDA modeling



where  $|\{d : t \in d\}|$  is the document count where 't' appears, and term–frequency function satisfying the condition  $tf(t, d) \neq 0$ . The term–frequency inverse document frequency  $tf - idf(t)$  is given

$$tf - idf(t) = tf(t, d) \cdot idf(t) \quad (7)$$

Using SST-LDA, each document is assigned a topic directly by measuring the similarity of the document. The documents can also be assigned topic based on the clusters. The documents are initially clustered. The documents are assigned to a topic and then the topic is assigned to the topic models. When a topic is closely related to several clusters, it is likely a noisy topic 'n.' Otherwise, it is a meaningful topic 'm.' The measure called cluster entropy (CE) is given as follows:

$$CE(t) = - \sum_{q \in Q} p(m|n) \log p(m|n) \quad (8)$$

The larger value of  $CE(t)$  the more likely the 't' is a noisy topic. The topic whose  $CE(t)$  is larger than a threshold (empirically set to 4.18) which is removed. After removing noisy topics, we obtained approximately 20 meaningful topics as the final set (Twitter users posts) used for our empirical comparison later. We quantitatively evaluated the efficiency of our SST-LDA model and compared with standard LDA model.

**Kullback–Leibler (KL) divergence** For our experiments, the KL divergence is chosen to obtain the similarity between original Twitter post A and sanitized Twitter post B for continuous distributions and invariant under parameter transformations. Given two probability distributions of two texts, A and B, the KL divergence measures the expected number of additional data required to code sections from A when using a code based on B. For probability distributions A and B of a discrete random variable i over topics, their KL divergence is defined as below:

$$\begin{aligned} D_{KL}(A||B) &= - \sum_i A(i) \log B(i) + \sum_i A(i) \log A(i) \\ &= \sum_i A(i) \log \frac{A(i)}{B(i)} \\ D_{KL}(A||B) &= H(A, B) - H(A) \end{aligned} \quad (9)$$

where  $H(A, B)$  is the cross-entropy of A and B, and  $dH(A)$  is the entropy of A. The KL divergence is asymmetric, and the real distance function is calculated as below:

$$D_{SST}(A, B) = \frac{1}{2} (D_{KL}(A||B) + D_{KL}(B||A)) \quad (10)$$

The  $D_{SST}$  is symmetric real distance function although it reserves the benefits of KL divergence. As the SST-LDA model obtain the information entropy from topic

distribution, sensitive topic, and access rule, our SST-LDA model empirically quantifies the dissimilarity of social messages (A and B) over these three types of unique attributes.

## 4 Experimental Results

The implementation progressed with the relationship calculation of Twitter users and the semantic analysis of the information shared by the users. The NodeXL<sup>1</sup> crawled up to 11,000 followers and friends of 790 Twitter users. The active news feeds collected using Twitter 4 J<sup>2</sup> API are stored as 157 chunked document files. The semantic analysis of the message shared by the user is analyzed using the Stanford NLP<sup>3</sup> API for POS tagging. It is the Java implementation of a named entity recognizer. The query processing is enabled using Apache Jena<sup>4</sup> API which provides the query engine for DBpedia knowledge base using SPARQL. The RDF serializations which Jena supports as both input and output formats are subjected to run on Hadoop MapReduce function for clustering. The SPARQL query is written to obtain all resources containing the topic which matches with the meaning of the sensitive term in the document. All the taxonomic categories of the sensitive word are represented as a hierarchy of generalized words. For the sensitive term Malaria, a query is written for retrieving all resources containing the term in them. The relationship strength between the users is calculated for the identified topic which is 'Health.' Its taxonomic categories are retrieved and annotated with the word accordingly shown in Fig. 3.

The annotated sensitive words are stored in the database with respective annotations. From Fig. 4, it is observed that the proposed model increases linearly with the number of documents with access rule generator and compared without access rule generator. The performance of the annotation process has greatly improved, and the difficulties with slow processing of the lengthy information are also reduced drastically using rule generator. The accuracy of proposed model increases for certain topic clusters, which is shown in Fig. 5. The maximum accuracy of 94% is achieved in identifying the topic semantically, and six sensitive topic clusters are identified in our experiments. The topic distribution and topic sensitivity score of 16 topics are shown in Fig. 6, and it is observed that the topics with more sensitive score give better accuracy on topic distribution. The accuracy of our SST-LDA model is calculated and compared with standard LDA model, which is shown in Fig. 7.

---

<sup>1</sup><https://nodexl.codeplex.com/>.

<sup>2</sup><http://twitter4j.org/en/>.

<sup>3</sup><http://nlp.stanford.edu/>.

<sup>4</sup><https://jena.apache.org/>.

Social ID of User Sharing the information  
421015442354

Information Shared:  
I have been suffering malaria for a week now #DocHelp

User to view information  
146554568878

Identified Topic : Health

Relationship Strength : 1.4578787

Access Policy : <health, friend, private>

Sanitized Message:  
I have been suffering from protozoan for a week now #DocHelp

Fig. 3 SST-sanitization for ‘Health’ content

Fig. 4 Access rule generator performance

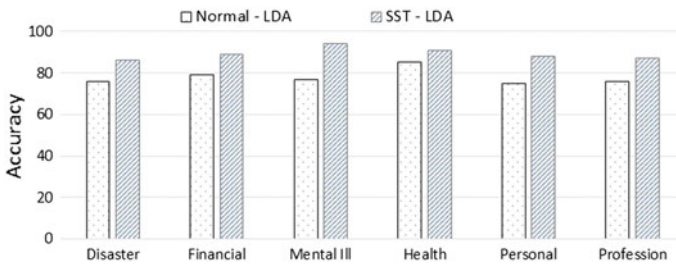
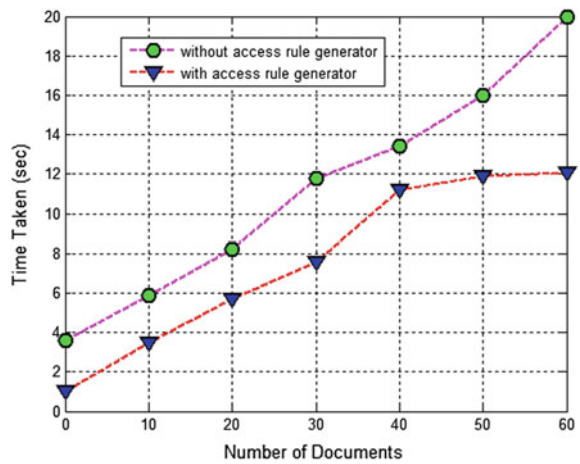


Fig. 5 Sensitive topic clusters—accuracy

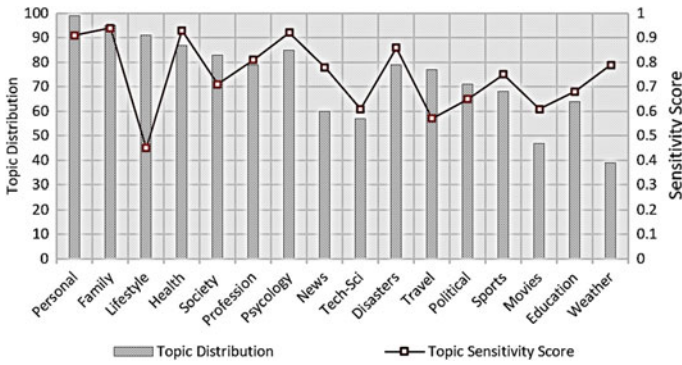
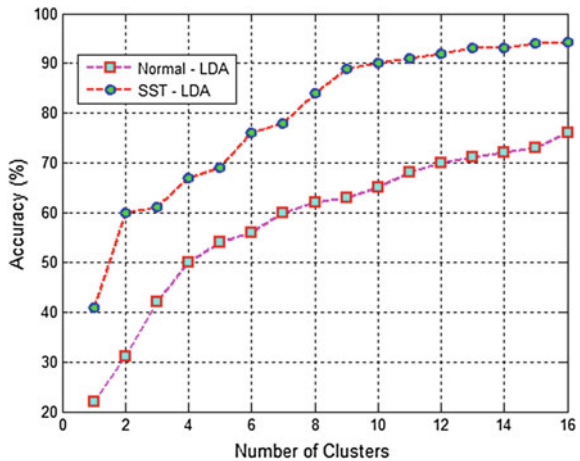


Fig. 6 Normal-LDA versus SST-LDA

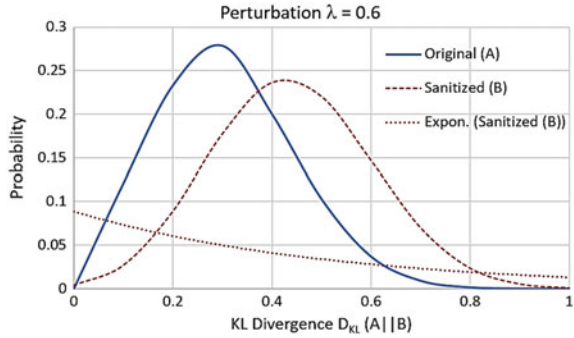
Fig. 7 Topic distribution and sensitivity score



We randomly considered 100 probability distributions on the 16 topic clusters and observed at the KL divergence with perturbation  $\lambda = 0.6$  is shown in Fig. 8. The two distributions of original post  $P(A)$  and sanitized post  $P(B)$  showing the real distance function  $D_{KL}(A||B)$  where the sanitized post  $B$  gives an indication of post been modified with small deviations from the original post  $A$  by maintaining the actual semantic of the sensitive post. The average case for information loss that falling between the two distributions is well described by the least informative prior exponentially.

The sensitive topic clusters are identified as {Disaster, Financial, Mental Illness, General Health, Personal, and Professional} is given in Table 1. The Cluster Entropy  $CE(t)$  is computed along with the similarity score of topics. Based on the entropy values, the sensitivity of the document is categorized as {very low, low, high, and very high}. The access policy which is identified for each sensitive topic

**Fig. 8** KL—divergence  $D_{KL}(A||B)$



**Table 1** Empirical values of sensitive topics identified

Topic	#docs identified	Cluster entropy	Sensitivity level	Similarity score (t)	Access policy referral	Automatic sanitization
Disaster	45	2.48	High	1.45	Yes	No
Financial	22	1.48	Very high	0.57	Strictly yes	Strictly yes
Mental Ill	16	0.45	Very high	0.09	Strictly yes	Strictly yes
Health	32	1.21	High	0.17	Yes	Yes
Personal	17	1.97	Very high	0.74	Strictly yes	Yes
Profession	19	3.11	High	0.84	Strictly yes	Yes

and it is recommended for automatic sanitization. To ensure the sensitivity of the information which is truly detected, the relationship strength of users on the selective topic is crowdsourced.

## 5 Conclusion

In this paper, we empirically sanitized the Twitter users post using SST-LDA model and compared with standard LDA, focusing on the variations between the two models. The proposed SST-LDA model is uniquely designed for the most sensitive Twitter posts related to six topics and showed its performance with a maximum accuracy of 94% in identifying the sensitive topic. Our observed comparison confirmed some previous empirical results and also revealed new findings based on accounting the topic sensitivity and cluster entropy. Therefore, it leads to a better and a more feasible mechanism to provide information privacy to the unaware users. In particular, we find the social media posts need precise sanitization and advanced access policy recommendations to the privileged user and sanitized information for the unprivileged user. As a part of the future work, we will extend

our study on how to summarize and visualize Twitter sanitization in a supervised deep learning approach for multilinguistic user profiles on a wide variety of sensitive topics.

## References

1. Li, K., Lin, Z., Wang, X.: An empirical analysis of users privacy disclosure behaviours on social network sites. *Int. J. Inform. Manag.* **52**(7) (2015)
2. Criado, N., Jose, M.: Such implicit contextual integrity in online social networks. *J. Inform. Sci.* **325**, 48–69 (2015)
3. Villata, S., Costabello, L., Delaforge, N., Gandon, F.: A social semantic web access control model. *J. Data Semant.* **2**, 21–36 (2013)
4. Carbunar, B., Rahman, M., Pissinou, N.: A survey of privacy vulnerabilities and defenses in geosocial networks. *IEEE Commun. Mag.* **51**(11) (2013)
5. Kandadai, V., Yang, H., Jiang, L., Yang, C.C., Fleisher, L., Winston, F.K.: Measuring health information dissemination and identifying target interest communities on twitter. *JMIR Res Protocols* **5**(2) (2016)
6. Imran-Daud, M., Sánchez, D., Viejo, A.: Privacy-driven access control in social networks by means of automatic semantic annotation. *Comput. Commun.* **76**, 12–25 (2016)
7. Ranjbar, A., Maheswaran, M.: Using community structure to control information sharing in OSN. *Comput. Commun.* **41**, 11–21 (2014)
8. Sánchez, D., Batet, M., Alexandre, V.: Utility-preserving sanitization of semantically correlated terms in textual documents. *J. Inf. Sci.* **279** (2014)
9. Sánchez, D., Batet, M., Viejo, A.: Automatic general-purpose sanitization of textual documents. *IEEE Trans. Inf. Forensics Security* **8**, 853–862 (2013)

# A Random Fourier Features based Streaming Algorithm for Anomaly Detection in Large Datasets



Deena P. Francis and Kumudha Raimond

**Abstract** Anomaly detection is an important problem in real-world applications. It is particularly challenging in the streaming data setting where it is infeasible to store the entire data in order to apply some algorithm. Many methods for identifying anomalies from data have been proposed in the past. The method of detecting anomalies based on a low-rank approximation of the input data that are non-anomalous using matrix sketching has shown to have low time, space requirements, and good empirical performance. However, this method fails to capture the non-linearities in the data. In this work, a kernel-based anomaly detection method is proposed which transforms the data to the kernel space using random Fourier features (RFF). When compared to the previous methods, the proposed approach attains significant empirical performance improvement in datasets with large number of examples.

**Keywords** Streaming data · Anomaly detection · Random Fourier features  
Matrix sketching

## 1 Introduction

Large data are encountered in many real-world applications. Due to the nature of this data, storing and processing of such data as a whole become infeasible. One of the important problems in modern applications is detecting anomalies. An anomaly is a datapoint that does not conform to the same pattern as the other data points

---

D. P. Francis (✉) · K. Raimond  
Department of Computer Sciences Technology, Karunya University,  
Coimbatore, Tamil Nadu, India  
e-mail: deena.francis@gmail.com

K. Raimond  
e-mail: kramond@karunya.edu

© Springer Nature Singapore Pte Ltd. 2018  
E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_18](https://doi.org/10.1007/978-981-10-7200-0_18)



in a dataset [1]. Detecting anomalies has become important in areas such as spacecraft systems [2], medicine, and finance [1]. Many approaches for anomaly detection have been proposed in the past. Subspace-based anomaly detection has been used by some works [3–5]. It involves computing a low-rank approximation of the non-anomalous input data points and then projecting the newly arrived points onto it. The anomalous points are discovered, and the non-anomalous points are used to update the low-rank approximation matrix. Huang et al. [6] used a matrix sketching technique for detecting anomalies in streaming data. They proposed a deterministic technique (DetAnom) which achieved better empirical results when compared to other scalable anomaly detection algorithms such as support vector machine (SVM) with linear as well as radial basis function (RBF) kernel, isolation forest [7], mass estimation [8], and unconstrained least-squares importance fitting [9]. They also achieved significant savings in time as well as space requirements. However, due to the nonlinearities in data encountered in modern applications, a linear subspace method like [6] fails to capture the behavior of the data. A kernel function maps the data to a non-linear feature space. Since directly applying kernel functions are computationally expensive, RFF method [10] is used to approximate the kernel function. In this work, RFF method [10] is used to transform the data to a feature space, and then the anomalies are identified.

This work is organized as follows. The notations used are described in Sect. 2. The previous related works are described in Sect. 3. The proposed approach is described in Sect. 4. The experimental results and discussion are provided in Sect. 5, and the conclusion is provided in Sect. 6.

## 2 Preliminaries

For a data matrix  $\mathbf{X} \in \mathbb{R}^{d \times n}$ ,  $n$  is the number of examples and  $d$  is the number of attributes of  $\mathbf{X}$ .  $\mathbb{I}_d$  is the identity matrix of size  $d \times d$ . The singular value decomposition (SVD) of  $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , where  $\mathbf{U} \in \mathbb{R}^{d \times d}$  is an orthogonal matrix,  $\mathbf{V} \in \mathbb{R}^{n \times n}$  is an orthogonal matrix, and  $\mathbf{\Sigma} \in \mathbb{R}^{d \times n} = \{\sigma_i\}$  is a diagonal matrix. The matrix  $\mathbf{\Sigma}$  contains the singular values of  $\mathbf{X}$ , sorted in the decreasing order, i.e.,  $\sigma_i \geq \sigma_j$  for  $i \leq j$ . The data arrives in a streaming fashion.  $\mathbf{X}_t$  denotes the data that arrives at time  $t$ , where  $\mathbf{X}_t \in \mathbb{R}^{d \times n_t}$ ,  $d$  is the number of attributes of the data, and  $n_t$  is the number of instances of the data at time  $t$ . The matrix  $\mathbf{X}_{[t]} \in \mathbb{R}^{d \times n_{[t]}}$  denotes the horizontal concatenation of matrices  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t \in \mathbb{R}^{d \times n_i}$ , where  $i = 1, \dots, t$ . The set of non-anomalous points identified at time  $t - 1$  is denoted by  $\mathbf{N}_{[t-1]}$ . The rank- $k$  approximation of  $\mathbf{N}_{[t-1]}$  is  $\text{SVD}_k(\mathbf{N}_{[t-1]_k}) = \mathbf{U}_{(t-1)_k} \mathbf{\Sigma}_{(t-1)_k} \mathbf{V}_{(t-1)_k}^T$ .

### 3 Previous Works

Density-based methods were used by [11, 12] to detect anomalies. The problem with this approach is that the all pairwise distance computation is expensive and hence cannot be used in large data. Nicolas and McDermott [13] proposed an autoencoder and density estimation method which also has the problem of expensive computation. Many subspace based anomaly detection approaches have been proposed in the past. Such methods construct a low-rank subspace of the non-anomalous data points in order to detect anomalies. Huang et al. [4, 5] used a principal component analysis (PCA)-based method using a sliding window scheme. These approaches suffer from the drawback of poor scalability. In order to overcome the problem of scalability, both deterministic and randomized matrix sketching-based techniques were proposed by [6]. The deterministic method (DetAnom) is based on the frequent directions (FD) algorithm of [14]. In their method, the rank- $k$  approximation of the non-anomalous points observed at time  $t - 1$ ,  $\mathbf{N}_{(t-1)}$  is computed. Using its left singular vectors  $\mathbf{U}_{(t-1)_k}$ , the anomaly score of a new data point  $\mathbf{x}_i$  is constructed as follows.

$$a_i = \|(\mathbb{I}_d - \mathbf{U}_{(t-1)_k} \mathbf{U}_{(t-1)_k}^T) \mathbf{x}_i\| \quad (1)$$

The points that have anomaly score greater than a threshold are marked as anomalies, and the rest are marked as non-anomalies. The left singular vectors are updated with the newly discovered non-anomalous points using a modified FD algorithm. This algorithm like most of the previous works does not capture the non-linearities in the data. Kernel-based data transformation can be used to overcome the drawback of the previous methods.

### 4 Proposed Approach

The proposed algorithm first uses RFF method [10], and then applies the FD-based anomaly detection algorithm, **DetAnom** of [6]. The proposed algorithm, **RFFAnom**, is shown in Algorithm 1.  $\mathbf{X}_{(t-1)}$  is the set of data points at time  $(t - 1)$ , and  $\mathbf{X}_t$  is the set of points at time  $t$  (new points). The algorithm starts with an initial set of non-anomalous points  $\mathbf{N}_{t-1}$  using which an initial sketch matrix  $\mathbf{B}_{t-1}$  and the matrix  $\mathbf{U}_{(t-1)_k}$  are computed. The columns of  $\mathbf{X}_{t-1}$  are made to have unit  $l-2$  norm, obtained by normalizing  $\mathbf{X}_{t-1}$ . As in [6], it is assumed that at any time  $t$ , a set of new points  $\mathbf{X}_t$  arrives. This batch (set of points)  $\mathbf{X}_t$  is transformed to the kernel space using the *FeatureMap* function in the Algorithm 2. Here,  $m$  is the number of feature maps to be generated. In this work,  $m$  is set to be equal to  $d$ , as the aim was not to perform dimensionality reduction, but rather to obtain a better representation of the non-anomalous points. The transformed points,  $\mathbf{Y}_t \in \mathbb{R}^{m \times n}$ , are obtained as a consequence of applying the *FeatureMap* function. These points are also normalized in order to make its columns to have unit  $l-2$  norm. The anomaly scores  $a_i$  are calculated as the distance between the points  $\mathbf{y}_i$  in  $\mathbf{Y}_t$  and the projection of the points  $\mathbf{y}_i$  onto

**Algorithm 1** RFFAnom

---

**Input:**  $\mathbf{X}_t \in \mathbb{R}^{d \times n_t}$ ,  $\mathbf{U}_{(t-1)k} \in \mathbb{R}^{d \times k}$ ,  $\eta \in \mathbb{R}$ ,  $\mathbf{B}_{t-1} \in \mathbb{d}^{m \times l}$ ,  $\mathbf{N}_t \leftarrow []$ ,  $\mathbf{A}_t \leftarrow []$ ,  $\zeta \in \mathbb{R}$   
Initial  $\mathbf{N}_{t-1}$  is used to compute  $\mathbf{B}_{t-1}$   
**for** each new set of points  $\mathbf{X}_t$  **do**  
   $\mathbf{Y}_t = \text{FeatureMap}(\mathbf{X}_t, \zeta)$   
  **for** each point  $\mathbf{y}_i$  in  $\mathbf{Y}_t$  **do**  
     $a_i = \|(\mathbb{1}_d - \mathbf{U}_{(t-1)k} \mathbf{U}_{(t-1)k}^T) \mathbf{y}_i\|$   
    **if**  $a_i \leq \eta$  **then**  
       $\mathbf{N}_t \leftarrow [\mathbf{N}_t, \mathbf{y}_i]$   
    **end if**  
  **end for**  
   $\mathbf{N}_{[t]} \leftarrow [\mathbf{N}_{[t-1]}, \mathbf{N}_t]$   
   $\mathbf{B}_t \leftarrow [\mathbf{B}_{t-1}, \mathbf{N}_{[t]}]$   
   $\tilde{\mathbf{U}}_t \tilde{\Sigma}_t \tilde{\mathbf{V}}_t^T \leftarrow \text{SVD}_1(\mathbf{B}_t)$   
   $\mathbf{B}_t \leftarrow \tilde{\mathbf{U}}_t \text{diag}(\sqrt{\tilde{\sigma}_{t_1}^2 - \tilde{\sigma}_{t_1}^2}, \dots, \sqrt{\tilde{\sigma}_{t_{l-1}}^2 - \tilde{\sigma}_{t_{l-1}}^2}, 0)$   
   $\tilde{\mathbf{U}}_{t_k} \leftarrow [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$   
**end for**  
**return**  $\mathbf{B}_t$  and  $\tilde{\mathbf{U}}_{t_k}$

---

the rank- $k$  subspace  $\mathbf{U}_{(t-1)k}$  of the non-anomalous points  $\mathbf{N}_{t-1}$ . If  $a_i$  is smaller than a threshold  $\eta$ , then the corresponding point is appended to the set of non-anomalous points  $\mathbf{N}_t$ . After all the non-anomalous points are obtained, the left singular vectors  $\mathbf{U}_{(t)}$  are updated by using the FD algorithm. In this part of the algorithm, the sketch matrix  $\mathbf{B}_t \in \mathbb{R}^{m \times l}$  is updated with the new set of non-anomalous points  $\mathbf{N}_t$ . Here,  $l$  is set as  $\sqrt{m}$  as suggested by [6]. Finally, the new set of left singular vectors  $\tilde{\mathbf{U}}_{t_k}$  is obtained. A diagram describing the proposed method is shown in Fig. 1. The run-

**Algorithm 2** Feature Map( $\mathbf{X}_t, \zeta$ )

---

$\mathbf{R} \leftarrow$  generate Gaussian random matrix with standard deviation  $\zeta$ ,  $\mathbf{R} \in \mathbb{R}^{m \times d}$   
 $\gamma \leftarrow$  Sampled uniformly at random from  $[0, 2\pi]$ ,  $\gamma \in \mathbb{R}^{m \times 1}$   
 $\mathbf{Y}_t \leftarrow \sqrt{\frac{2}{m}} \cos(\mathbf{R}\mathbf{X}_t + \gamma)$   
**return**  $\mathbf{Y}_t$

---

ning time of *DetAnom* algorithm is  $O(\max\{dn_t, l, dl^2\})$ , and the proposed algorithm is slower by a factor of  $\sqrt{d}$ . This does not affect the running time in the experiments to a great extent because the datasets considered do not have high dimensionality. By using RFF, the running time of applying kernel functions is reduced significantly. The space required by the algorithm is  $O(d \cdot \max_t \{n_t\} + dl)$ .

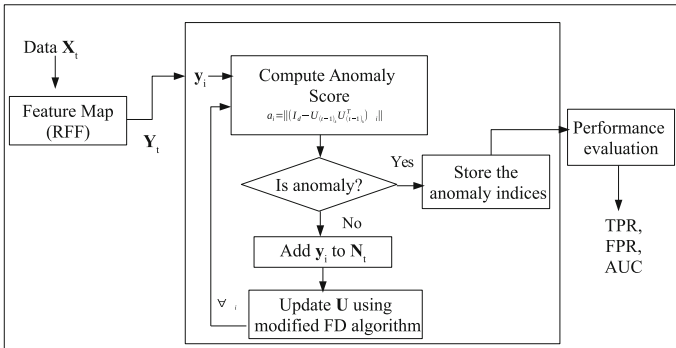


Fig. 1 Proposed method for anomaly detection from a set of new input points

## 5 Experimental Results

All experiments were carried out in a Linux machine with 3.5 GHz Intel Core i7 processor and 16 GB of RAM. For all the experiments, the proposed algorithm **RFFAnom** is compared against deterministic algorithm **DetAnom** of [6]. The *DetAnom* algorithm has been shown to have better empirical performance than many other scalable anomaly detection algorithms [6]. Here, non-anomalous points are labeled as 0 and the anomalous points are labeled as 1. From the set of non-anomalous data points, 2000 points are drawn at random and they comprise the initial set of non-anomalous points. The size of the data arriving as input to the algorithm at each time  $t$  is set as 5000 as suggested by [6].

### 5.1 Datasets Used

- COD-RNA [15]: contains 488,565 genome sequences with eight attributes. The anomalies in this case are the set of non-coding RNAs. The number of examples in classes 0 and 1 are 325710 and 162855, respectively, and the percentage of anomalies is 33%.
- Forest [16]: contains 286048 instances of forest cover types. The data were obtained from <http://odds.cs.stonybrook.edu/forestcovercovertype-dataset/> and contained 10 attributes. The number of examples in classes 0 and 1 are 283301 and 2747, respectively, so the dataset has 0.9% anomalies.
- Protein-Homology [17]: contains 145751 instances and 74 attributes. The number of class 0 and 1 instances are 144455 and 1296 respectively. It has 0.8% anomalies.
- Shuttle: contains nine attributes and 49095 instances out of which 3511 are outliers. The number of non-anomalous points is 45586, so the percentage of anomalies is 7%. The data were obtained from <http://odds.cs.stonybrook.edu/shuttle-dataset/>.

- MNIST [18]: contains a total of 7603 instances and 100 attributes. The number of class 0 and 1 instances is 6903 and 700, respectively. It has 9% anomalies. The data were obtained from <http://odds.cs.stonybrook.edu/mnist-dataset/>.
- HTTP: contains 41 attributes and 200000 instances. The number of class 0 and 1 instances is 160555 and 39445, respectively, and it has 19% anomalies. The data were obtained from UCI repository [19].

## 5.2 Performance Metrics

The metrics used to evaluate the result of the algorithm are described below. *True Positive Rate (TPR)*: It is the proportion of correctly identified instances. Here, it is the proportion of anomalies that have been correctly identified.

$$TPR = \frac{TP}{(TP + FN)} \quad (2)$$

*False Positive Rate (FPR)*: It is the proportion of negative instances that have been correctly identified. Here, it is the proportion of non-anomalous points that have been correctly identified.

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

*Area Under the Curve (AUC)*: It is a metric computed from the plot of TPR and FPR. If this value is close to 1, then the performance of the algorithm is good, and if it is less than 0.5, the performance is poor.

## 5.3 Results

The receiver operating characteristic (ROC) plots of the algorithms *DetAnom* and *RFFAnom* are shown in the Figs. 2, 3 and 4. For the cod-RNA and Forest datasets, the proposed algorithm, *RFFAnom*, performs much better than *DetAnom*. In particular, *DetAnom* performs suboptimally for small values of FPR, whereas *RFFAnom* has better results. The AUC values and the time taken for each dataset are shown in Table 1.

In Fig. 3a, for the Protein-Homology dataset, *DetAnom* performs slightly better than *RFFAnom*. It can be seen from Fig. 4a that for the MNIST dataset, the proposed algorithm performs better than *DetAnom*. In Fig. 4b, for the HTTP dataset, the AUC value of the proposed algorithm is 0.995, which is significantly better than that of *DetAnom*. The figure also shows how well the proposed algorithm performs since

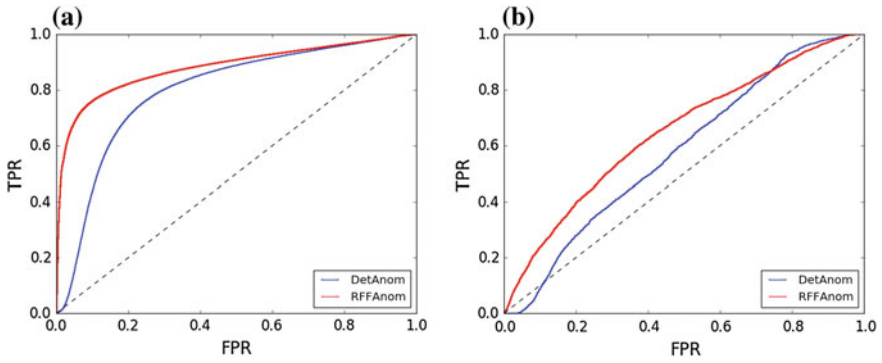


Fig. 2 ROC curves of RFFAnom and DetAnom algorithms for **a** cod-RNA (left), **b** Forest (right)

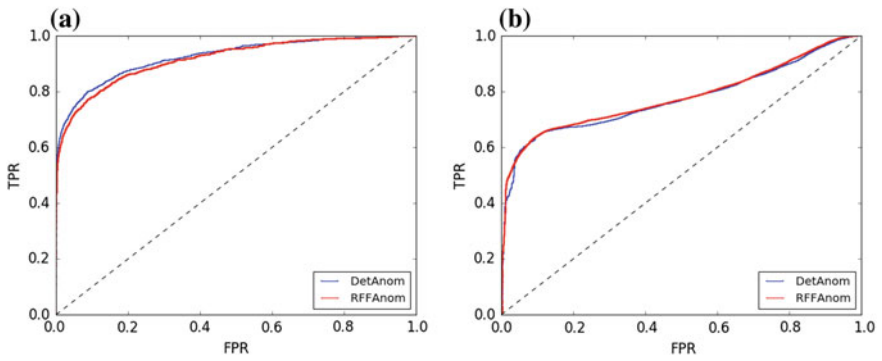


Fig. 3 ROC curves of RFFAnom and DetAnom algorithms for **a** Protein-Homology (left), **b** Shuttle (right)

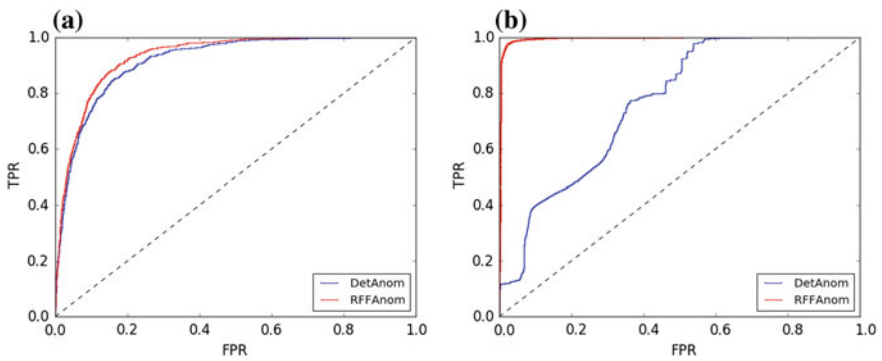


Fig. 4 ROC curves of RFFAnom and DetAnom algorithms for **a** MNIST (left), **b** HTTP (right)

**Table 1** AUC values and time taken by DetAnom and RFFAnom algorithms for various datasets

Dataset	Algorithm	AUC	Time taken (s)
COD-RNA	DetAnom	0.797620	3.3825
	RFFAnom	<b>0.883272</b>	3.3717
Forest	DetAnom	0.581079	2.0930
	RFFAnom	<b>0.651525</b>	2.2101
Protein-Homology	DetAnom	<b>0.924820</b>	5.6235
	RFFAnom	0.917973	6.1562
Shuttle	DetAnom	0.773587	0.3910
	RFFAnom	<b>0.781500</b>	0.4075
MNIST	DetAnom	0.917452	0.6308
	RFFAnom	<b>0.933240</b>	0.7404
HTTP	DetAnom	0.764248	4.0326
	RFFAnom	<b>0.995234</b>	4.4200

its graph lies close to the y-axis. In general, the proposed approach performs much better than *DetAnom* for datasets with large number of instances. The results indicate that the feature space transformation improves the anomaly detection capability of the proposed algorithm. Many datasets that are available today have some kind of non-linearity present. The kernel feature space transformation (RFF) used in this work effectively exploits this nature of the data.

## 6 Conclusion

Detecting anomalies from streaming data is an important application in many areas. In the past, many methods for identifying anomalies from data have been proposed. But most of these algorithms suffer from the problem of poor scalability. In this work, a RFF-based anomaly detection method is proposed. It makes use of a kernel feature space transformation of the data points and a FD-based anomaly detection scheme. The proposed method has a low running time and space requirements and is hence applicable to large datasets. Empirical results indicate that a significant improvement in the performance was obtained for large datasets when compared to the previous method.

**Acknowledgements** The authors would like to thank the financial support offered by the Ministry of Electronics and Information Technology (MeitY), Govt. of India under the Visvesvaraya Ph.D Scheme for Electronics and Information Technology.

## References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. *ACM Comput. Surv. (CSUR)* **41**(3) (2009). <https://doi.org/10.1145/1541880.1541882>
2. Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 401–410. ACM (2005). <https://doi.org/10.1145/1081870.1081917>
3. Lakhina, A., Crovella, M., Diot, C.: Characterization of network-wide anomalies in traffic flows. In: *SIGCOMM* (2004). <https://doi.org/10.1145/1028788.1028813>
4. Huang, L., Nguyen, X., Garofalakis, M., Jordan, M.I., Joseph, A., Taft, N.: In-network PCA and anomaly detection. In: *NIPS*, pp. 617–624 (2006)
5. Huang, L., Nguyen, X., Garofalakis, M., Hellerstein, J.M., Jordan, M.I., Joseph, A.D., Taft, N.: Communication-efficient online detection of network-wide anomalies. In: *INFOCOM* (2007). <https://doi.org/10.1109/INFCOM.2007.24>
6. Huang, H., Kasiviswanathan, S.P.: Streaming anomaly detection using randomized matrix sketching. *Proc. VLDB Endow.* **9**(3), 192–203 (2015)
7. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: *IEEE ICDM*, pp. 413–422 (2008). <https://doi.org/10.1109/ICDM.2008.17>
8. Ting, K.M., Zhou, G.T., Liu, F.T., Tan, J.S.: Mass estimation and its applications. In: *ACM SIGKDD* (2010). <https://doi.org/10.1145/1835804.1835929>
9. Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., Kanamori, T.: Statistical outlier detection using direct density ratio estimation. *KAIS* **26**(2) (2011)
10. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: *Advances in Neural Information Processing Systems*, pp. 1177–1184 (2007)
11. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: *ACM Sigmod Record*, vol. 29, pp. 93–104. ACM (2000). <https://doi.org/10.1145/342009.335388>
12. Tang, J., Chen, Z., Fu, A.W.C., Cheung, D.W.: Enhancing effectiveness of outlier detections for low density patterns. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535–548. Springer, Berlin, Heidelberg (2002). [https://doi.org/10.1007/3-540-47887-6\\_53](https://doi.org/10.1007/3-540-47887-6_53)
13. Nicolau, M., McDermott, J.: A hybrid autoencoder and density estimation model for anomaly detection. In: *International Conference on Parallel Problem Solving from Nature*, pp. 717–726. Springer International Publishing (2016). [https://doi.org/10.1007/978-3-319-45823-6\\_67](https://doi.org/10.1007/978-3-319-45823-6_67)
14. Liberty, E.: Simple and deterministic matrix sketching. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 581–588. ACM (2013). <https://doi.org/10.1145/2487575.2487623>
15. Uzilov, A.V., Keegan, J.M., Mathews, D.H.: Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. *BMC bioinform.* **7**(1) (2006)
16. Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. electron. agric.* **24**(3), 131–151 (1999)
17. Caruana, R., Joachims, T., Backstrom, L.: KDD-Cup 2004: results and analysis. *ACM SIGKDD Explor. Newslett.* **6**(2), 95–108 (2004)
18. Lecun, Y., Cortes, C.: The MNIST database of handwritten digits. (2009). <http://yann.lecun.com/exdb/mnist/>
19. UCI repository. <https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/> (1999)



# SBKMEDA: Sorting-Based K-Median Clustering Algorithm Using Multi-Machine Technique for Big Data



E. Mahima Jane and E. George Dharma Prakash Raj

**Abstract** Big Data is the term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. Clustering is an essential tool for clustering Big Data. Multi-machine clustering technique is one of the very efficient methods used in the Big Data to mine and analyse the data for insights. K-Means partition-based clustering algorithm is one of the clustering algorithm used to cluster Big Data. One of the main disadvantage of K-Means clustering algorithms is the deficiency in randomly identifying the K number of clusters and centroids. This results in more number of iterations and increased execution times to arrive at the optimal centroid. Sorting-based K-Means clustering algorithm (SBKMA) using multi-machine technique is another method for analysing Big Data. In this method, the data is sorted first using Hadoop MapReduce and mean is taken as centroids. This paper proposes a new algorithm called as SBKMEDA: Sorting-based K-Median clustering algorithm using multi-machine technique for Big Data to sort the data and replace median with mean as centroid for better accuracy and speed in forming the cluster.

**Keywords** Big Data • Clustering • K-Means algorithm • Hadoop MapReduce • SBKMA

---

E. Mahima Jane (✉)

Department of Computer Application, Madras Christian College,  
Tambaram 600059, India  
e-mail: mahima.jane@gmail.com

E. George Dharma Prakash Raj

Department of Computer Science and Engineering, Bharathidasan University,  
Trichy 620023, India  
e-mail: georgeprakashraj@yahoo.com

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_19](https://doi.org/10.1007/978-981-10-7200-0_19)

## 1 Introduction

Big Data analytics examines large amount of data to uncover hidden patterns, correlations and other insights. Many social networking Websites such as Facebook, Twitter have billions of users who produce gigabytes of contents per minute. Similarly, many online retail stores conduct business worth millions of dollars. Hence, it is necessary to have efficient algorithms to analyse and group the data and derive meaningful information [1]. Clustering of data is a method by which large sets of data is grouped into clusters of smaller sets of similar data.

Clustering algorithms have emerged as an alternative powerful meta-learning tool to accurately analyse the massive volume of data generated by modern applications. Multi-machine techniques are flexible in scalability and offer faster response time to the users. K-Means clustering algorithm follows a simple and easy way to classify a given data set through a certain number of clusters fixed a priori. The main idea is to define k-centroids, one for each cluster. SBKMA: Sorting-based K-Means clustering algorithm using multi-machine techniques a partition-based clustering algorithm which reduces the iteration and execution time of traditional K-Means algorithm. It is found that SBKMA based on mean reduces the iteration and execution time and this gave an idea to include median instead of mean for this proposed work. This paper is further written as follows. It discusses other related and relevant work in Sect. 2. Section 3 gives a small introduction about multi-machine clustering technique followed by the proposed work in Sect. 4. The Experimentation and Analysis is explained in Sect. 5 followed by the Conclusion in Sect. 6.

## 2 Related Work

Jane and George Dharma Prakash Raj [1] proposes an Algorithm called as SBKMA. In the SBKMA algorithm, the centroids are identified by sorting the objects first and then identifying the mean from the partition done as per the K-clusters. Each K-cluster is partitioned and mean of each cluster is taken as centroid. Multi-machine clustering technique which is discussed in Sect. 3 allows to breakdown the huge amount of data into smaller pieces which can be loaded on different machines and then uses processing power of these machines to solve the problem. Hence, number of iterations and execution time are considerably reduced. Vrinda and Patil [2] describe the various K-Means algorithm, their advantages and disadvantages. Patil and Vaidya [3] present the implementation of K-Means clustering algorithm over a distributed environment using Apache Hadoop. This work explains the design of K-Means algorithm using Mapper and Reducer routines. It also provides the steps involved in the implementation and execution of the K-Means algorithm. Baswade and Nalwade [4] in this paper present the drawback of traditional K-Means algorithm of selection initial centroid is removed. Here,

mean is taken as the initial centroid value. Vishnupriya and Sagayaraj Francis [5], in this method, describe that the K-Means algorithm is used to cluster the data for different type of data sets in Hadoop framework and calculate the sum of squared error value for the given data. Gandhi and Srivastava [6] suggest different partitioning techniques, such as k-means, k-medoids and clarans. In terms of data set attributes, the objects within single clusters are of similar characteristics where the objects of different cluster have dissimilar characteristic. Bobade [7] describes the concept of Big Data along with Operational versus Analytical systems of Big Data.

### 3 Multi-Machine Clustering Technique

Multi-Machine clustering technique is a clustering method where data are analysed using multiple machines. It enables to reduce the execution time and improves the speed of the clusters formed. Multi-machine clustering technique is divided into parallel clustering and map-reduced based clustering. This type of techniques allows the data to be divided into multiple servers and processes the request.

### 4 Proposed Algorithm: SBKMEDA: Sorting-Based K-Median Clustering Algorithm Using Multi-Machine Technique

In SBKMA algorithm, sorting is initially done with multi-machine technique. K-value is chosen randomly. Mean is taken as the centroid. The proposed SBKMEDA algorithm is the modified version of the previous SBKMA algorithm suggested by us. Here, median is taken as centroids to get more accurate results in the formation of clusters.

The proposed SBKMEDA algorithm is given below:

- Step 1: Load the data set
- Step 2: Choose K-clusters.
- Step 3: Sort the data set
- Step 4: Calculate the median and choose centroids based on the K-clusters
- Step 5: Find the distance between the objects and centroids
- Step 6: Group objects with minimum distance
- Step 7: Repeat 4, 5 and 6 until no change in the pattern
- Step 8: Stop the program (Fig. 1).

In this algorithm, the data is sorted first using Hadoop MapReduce. When sorting is completed, the data is divided into k-clusters. Median is taken as centroids. Then, the distance between the object and the centroid is calculated. When

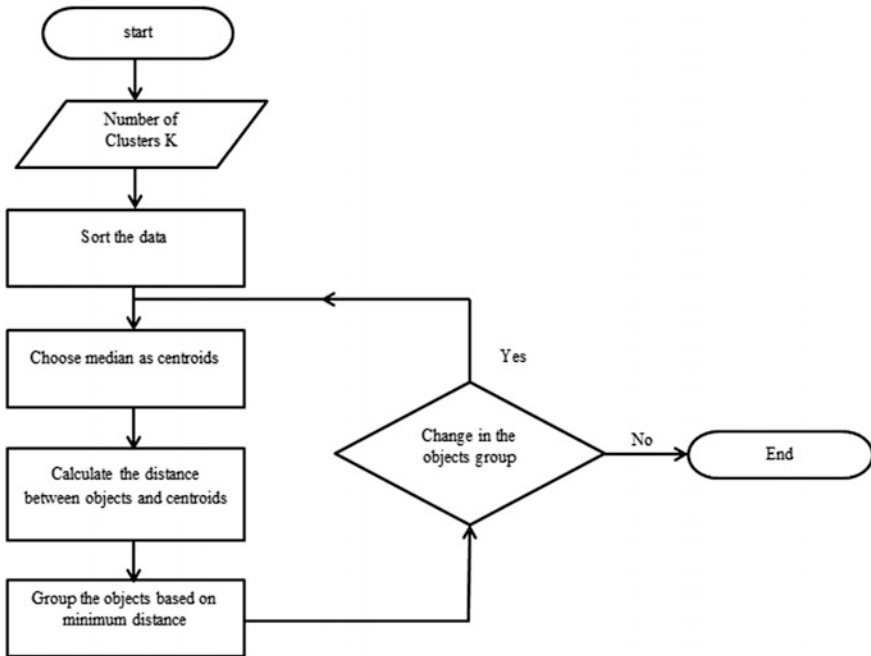


Fig. 1 Working of SBKMEDA

the distance is calculated, the objects are grouped with the nearest centroid. Repetitively the process is carried out till there is no change in the group. When there is no change in the formation, the process is finished.

## 5 Experimentation and Analysis

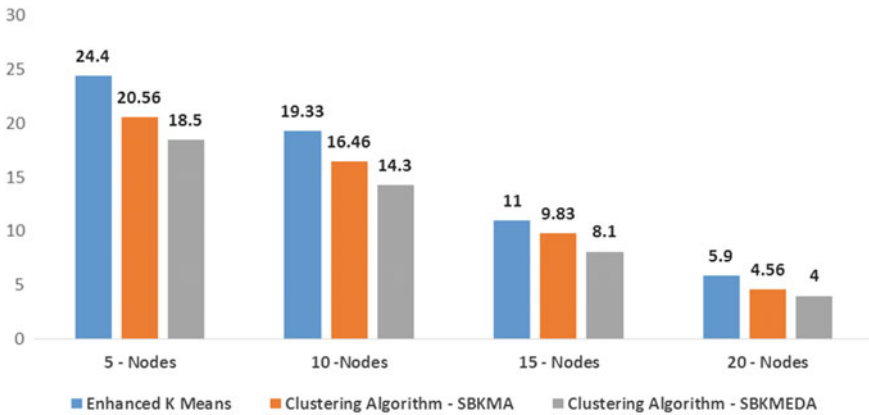
SBKMA algorithm, enhanced traditional K-Means algorithm [8] and the proposed SBKMEDA clustering algorithm are implemented in Hadoop MapReduce framework. Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data in parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner [9].

MapReduce usually splits the input data set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks [9].

The process of Experimentation of this paper using MapReduce using Multiple Machine Technique is explained next. First, Java file with 1 terabyte of mobile data set is randomly generated. This data is stored in the form of text files. This data is

**Table 1** Execution time

No. of nodes	Enhanced K-Means	Clustering algorithm—SBKMA	Clustering algorithm—SBKMEDA
5-nodes	24.4	20.56	18.5
10-nodes	19.33	16.46	14.3
15-nodes	11	9.83	8.1
20-nodes	5.9	4.56	4



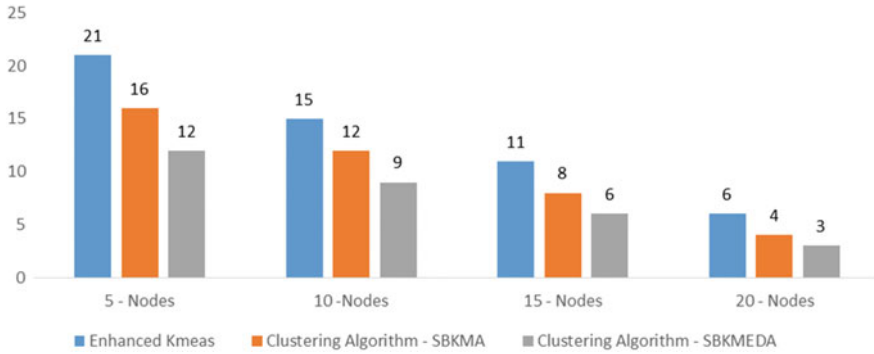
**Fig. 2** Execution time

consumed by the Hadoop MapReduce and the two algorithms are compared for analysis (Table 1).

The execution time of SBKMEDA is lesser when compared to SBKMA and the existing traditional enhanced K-Means algorithm. When the number of nodes is increased, the efficiency increases even more. This can be seen in Fig. 2. Execution time is measured in seconds. The number of nodes is different when compared to the previous SBKMA algorithm (Table 2).

**Table 2** No of iterations

No. of nodes	Enhanced K-Means	Clustering algorithm—SBKMA	Clustering algorithm—SBKMEDA
5-nodes	21	16	12
10-nodes	15	12	9
15-nodes	11	8	6
20-nodes	6	4	3



**Fig. 3** No. of iterations

Here, the number of times the algorithm is iterated is less for SBKMEDA when compared to SBKMA and the existing traditional enhanced K-Means algorithm as shown in Fig. 3.

## 6 Conclusion

In this paper, the algorithm is compared with already existing sorting-based K-Means algorithm and Enhanced traditional K-Means algorithm. Here, multi-machine technique with Hadoop MapReduce is used to sort the data. To improve the efficiency, median is taken as centroids. This algorithm reduces time and the number of iterations when compared with sorting-based K-Means algorithm. In future, it can be extended with other partition-based algorithms.

## References

1. Jane, M., George Dharma Prakash Raj, E.: SBKMA: sorting based K-Means clustering algorithm using multi machine technique for Big Data. *Int. J. Control Theory Appl.* **8**, 2105–2110 (2015)
2. Vrinda, Patil, S.: Efficient clustering of data using improved K-Means algorithm—a review. *Imp. J. Interdiscip. Res.* **2**(1) (2016)
3. Patil, Y.S., Vaidya, M.B.: K-Means clustering with MapReduce technique. *Int. J. Adv. Res. Comput. Commun. Eng.* (2015)
4. Baswade, A.M., Nalwade, P.S.: Selection of initial centroids for K-Means Algorithm. *IJCSMC* **2**(7), 161–164 (2013)
5. Vishnupriya, N., Sagayaraj Francis, F.: Data clustering using MapReduce for multidimensional datasets. *Int. Adv. Res. J. Sci. Eng. Technol.* (2015)
6. Gandhi, G., Srivastava, R.: Review paper: a comparative study on partitioning techniques of clustering algorithms. *Int. J. Comput. Appl.* (0975-8887) **87**(9) (2014)

7. Bobade, V.B.: Survey paper on Big Data and Hadoop. *Int. Res. J. Eng. Technol. (IRJET)* **03** (01) (2016)
8. Rauf, A., Sheeba, Mahfooz, S., Khusro, S., Javed, H.: Enhanced K-Mean clustering algorithm to reduce number of iterations and time complexity. *Middle-East J. Sci. Res.* **12**(7), 959–963 (2012)
9. [https://hadoop.apache.org/docs/r1.2.1/mapred\\_tutorial.html](https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)

# Cohesive Sub-network Mining in Protein Interaction Networks Using Score-Based Co-clustering with MapReduce Model (MR-CoC)



R. Gowri and R. Rathipriya

**Abstract** Nowadays, due to the data deluge situation, every computation has to be carried out in voluminous data. The sub-network mining from the complex and voluminous interaction data is one of the research challenges. The highly connected sub-networks will be more cohesive in the network. They are responsible for communication among the network, which is useful for studying their functionalities. A novel score-based co-clustering (MR-CoC) technique with MapReduce is proposed to mine the highly connected sub-network from interaction networks. The MapReduce environment is chosen to cope with complex, voluminous data and to parallelize the computation process. This approach is used to mine cliques, non-cliques, and overlapping sub-network patterns from the adjacency matrix of the network. The complexity of the proposed work is  $O(E_s + \log N_s)$ , which is minimal than the existing approaches like MCODE and spectral clustering.

**Keywords** MapReduce • Clustering • Protein interaction network  
Co-clustering • Functional coherence • Sub-network mining • Distributed computing • Big data

## 1 Introduction

The protein interaction networks (PIN) are one of the massive networks like social communication networks. It is one of the emerging big data in bioinformatics. The PIN can have both the dense part where the proteins are highly cohesive with one another and the sparse part where the communication will be minimal based on their connectivity [1, 2]. These cohesive sub-networks are responsible for any specific functionality like transcending the signals to various parts of the network.

---

R. Gowri (✉) · R. Rathipriya  
Department of Computer Science, Periyar University, Salem, Tamilnadu, India  
e-mail: gowri.candy@gmail.com

R. Rathipriya  
e-mail: rathi\_priyar@periyaruniversity.ac.in



They are called as cohesive subgroups. The disease proteins will infect these highly cohesive groups for affecting any cellular processes and spread the disease to various parts will be sooner than sparse parts. Data mining from the big data should require the appropriate computational setup to cope with the challenges of volume, variety, veracity, and velocity of the data.

The MapReduce framework invented by the Google has been the successful framework to handle the big data challenges by providing parallel processing facilities in the distributed environment [3–8].

Co-clustering is one of the data mining techniques for excavating the similar groups having cohesive characteristics. Co-clustering in MapReduce framework will be helpful to perform sub-network identification in the big data environment. It is the process of clustering the data matrix simultaneously on both the rows and the columns. The advantages of co-clustering over clustering [2, 9–11] are: can be applied to any two dimensions simultaneously; can derive the local model; and can group more similar patterns than the clustering process.

To measure the cohesiveness of the sub-networks, a novel score measure is proposed in this paper. The non-cliques may be densely connected/highly cohesive having more functional significance are ignored in many existing approaches. This score measure is used for co-clustering the adjacency matrix of the given network dataset. Many researchers negotiate that clustering the adjacency matrix of a network is meaningless. But the proposed approach overcomes the existing shortcoming, which is discussed in this paper. The available related researches, proposed methodology and its implementation along with their experimental setup, and the results are discussed in further sections.

## 2 Related Works

The literature study shows that there are many models were devised to mine the protein modules, protein patterns based on their topological properties. They used various graph-theoretic techniques, topological properties, and classification measures to confine their results. The study on related works is briefly confined in Table 1. The inference from this study is that the computational overhead and scalability issues can be avoided by parallelizing the computation in a distributed environment. The MapReduce programming model is chosen to overcome this issue. In many research articles [12, 13], the proposed algorithm finds either clique or other specific pattern with various size constraints. They have to be redefined to fit the further requirements. In case of MCODE algorithm [12], it suits for finding the clique, whereas it ignores many non-clique dense subgraphs. It does not perform well for a large dataset, enumerating all the subgraphs is an NP-hard problem. Its computational complexity is  $O(n^3)$ . In MapReduce-based clustering algorithms, many complex logics have to be carried out to perform subgraph mining. The hybridization of more than one graph mining algorithm will complicate the overall performance of the model.

**Table 1** Related research articles in the literature

Title	Algorithm	Measure	Input	Output
Detection of functional modules from protein interaction networks [14]	Clustering	Classification score	Weighted score	Network clusters
Identifying functional modules in protein–protein interaction networks: an integrated exact approach [15]	Mathematical optimization	–	Weighted vertex	Subgraphs
Protein interaction networks—more than mere modules [16]	Block method based on GO terms	Proposed error minimization	Adjacency matrix	Connection between connected and simple subgraphs
Weighted consensus clustering for identifying functional modules in protein–protein interaction networks [17]	Combines 4 clustering algorithms	Cluster coefficient	V, E	Network clusters
An automated method for finding molecular complexes in large protein interaction networks [11]	MCODE	Network density	Vertex, degree	Clique (connected subgraph)
A faster algorithm for detecting network motifs [18]	Enumerating subgraphs	Neighborhood	V, E, neighbor list	k-size subgraphs (network motifs)
Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs [19]	Edge sampling	–	Edge list and neighbor list	Subgraphs

### 3 Methodology: MR-CoC

#### 3.1 Network Representation and Terminology

Protein interaction network is represented using the undirected graph structure as  $G(V, E)$  [20–23]. Here, the proteins are the set of vertices or nodes ( $V$ ) whereas the interactions between these proteins are edges or links ( $E$ ) as in Fig. 1a. Edge is represented as  $E_i = \langle P_a, P_b \rangle$ ,  $P_a, P_b \in V$ . The PIN is an undirected graph ( $\langle P_a, P_b \rangle = \langle P_b, P_a \rangle$ ). The connectivity among the proteins is represented using the adjacency matrix, which is a square matrix of size  $|V| \times |V|$ . The adjacency matrix of the PIN represents the membership value of the edge as defined in the Eq. (1). The adjacency matrix ( $A$ ) of the undirected graph should be symmetric ( $A = A^T$ ).

$$A_{i,j} = \begin{cases} 1, & \text{Edge between } P_i \text{ and } P_j \\ 0, & \text{No Edge} \end{cases} \quad (1)$$

- Complete Graph: It has all possible edges of the graph [22, 24]. Every pair of vertices should have an edge except self-loops. All the entries of the adjacency matrix except the diagonal elements must be 1 as in Fig. 1b.
- Sub-network (or) Subgraph: The subgraph  $G_1(V_1, E_1)$  of a graph  $G(V, E)$  is a graph where  $V_1 \subset V$  and  $E_1 \subset E$  [21–23]. The adjacency matrix  $A_1$  of a subgraph  $G_1$  is a sub-matrix of  $A$ . In Fig. 1, the red circle represents the sub-network or subgraph.
- Clique: The clique is a complete subgraph that presents in a given graph  $G$  [22]. It possesses all the properties of the complete graph, but it is a subgraph.
- MapReduce Programming Model: The MapReduce is a programming paradigm, which simplifies large-scale data processing on commodity cluster by exploiting parallel map task and reduce task. It has emerged as the most popular computing framework for big data processing due to its simple programming model and automatic management of parallel execution [3, 5, 6, 25]

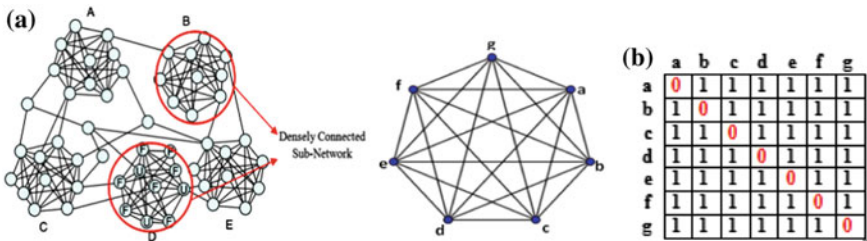


Fig. 1 Sample protein interaction network with sub-networks highlighted, a complete graph and its adjacency matrix

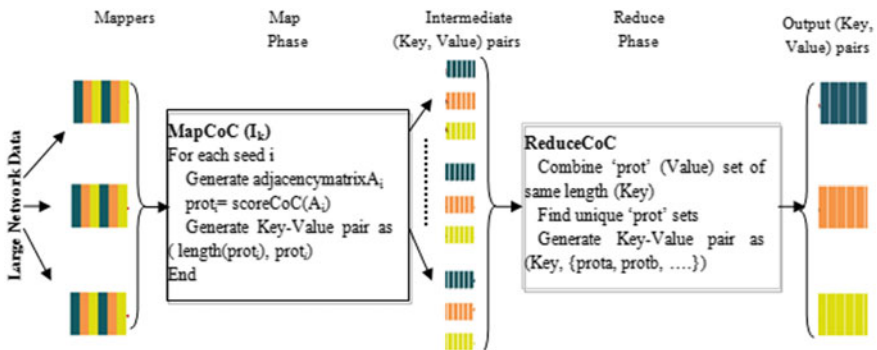


Fig. 2 Overview of the proposed MR-CoC model

### 3.2 Score-Based Co-clustering with MapReduce Approach (MR-CoC)

The score measure is used for co-clustering the PIN. The score is defined using the frequency of '1' in the adjacency matrix. The score of the adjacency matrix of the graph or subgraph can be calculated to access the nature of the derived local pattern from the given PIN using the equation. It is the ratio of the frequency of '1's to the number of elements as in the equation.

$$\text{score}(A_n) = \frac{\sum_{i=1, i \neq j}^n \text{freq}(a_{i,j})}{n(n-1)} \quad (2)$$

where  $A_n$  is the adjacency matrix of order 'n' with membership value (either '0' or '1') of the edge, the numerator represents the frequency of '1's in the adjacency matrix except the diagonal elements. The score value of the cliques and the complete graphs will always 1. The decimal values ranging below 1 can also be easily considered for mining non-cliques or dense sub-networks. The proposed approach, score-based co-clustering approach (Fig. 2), has the following steps:

1. Adjacency Matrix Generation.
2. Seed Generation: a set of initial random proteins to extract the sub-network.
3. Highly Connected Sub-Network (Clique): Mining using MR-CoC
4. Biological Significance of Modules using GO term finder.

#### Function scoreCoC( $A_n$ )

```

while (score < thres)
  Evaluate score( $A_n$ )
  Remove protein with low score (both row and column) from  $A_n$ 
end
prot=protein ids in  $A_n$ 
return prot
end

```

The MR-CoC has two main phases: map and reduce. The generated seeds are written in the text files, and they are fed as input to the map phase. For each seed generate the adjacency matrix by extracting the corresponding rows and columns of seed proteins from  $A$  as sub-matrix then follow the score-based co-clustering process as given in the algorithm. Interaction network is represented using the graph data structure. The complexity of the proposed work is  $O(E_s + \log N_s)$  where  $E_s$  is a number of edges in a seed and  $N_s$  represents the number of nodes in a seed. It is minimum than the MCODE ( $O(n^3)$ ) algorithm, which is widely used for mining subgraphs.

## 4 Experimental Setup

The Homo sapiens protein interaction dataset is initially chosen to attempt this methodology. The protein interaction networks are taken from string database [26], and their descriptions are given in Table 2. The seeds are an initial set of proteins for generating each subgraph, which will be redundant. The score measure is used to find the subgraph from each seed. The distinct subgraphs are extracted in the reduce phase. The proposed approach is implemented using MapReduce model in the Matlab. The environmental setup is discussed in the table. The workers represent the number of parallel threads chosen for this implementation.

## 5 Results and Discussion

The cohesive sub-networks are mined from the given protein interaction network using MR-CoC. Some of the cliques and non-cliques obtained using the proposed methodology are listed in Table 4. The number of cliques, non-cliques for different initial seed setup is given in Table 3, which depends on the initial seed selection.



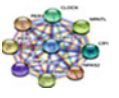
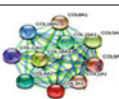

**Table 2** Experimental setup

Parameter	Value
Environment	Matlab 2015b (MapReduce Model)
Number of interactions	85,48,003
Number of proteins	19, 427
Number of workers (parallel threads)	10 workers
Number of seeds (max)	10 billion seeds
Seed length	50 proteins
Number of mappers	Environment-dependent
Minimum size of sub-network	5 proteins
Score value—minimum threshold (for non-cliques)	0.8

**Table 3** Overall computational result of proposed methodology MR-CoC

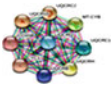
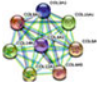



Number of seeds	Cliques	Non-cliques	Time (s)
100000	683	2766	1326.776033
500000	812	3178	1711.162498
1000000	2381	2190	8933.431215
5000000	1782	4822	12348.823441
10000000	2910	8821	33601.989910
50000000	4987	6975	95678.8965
100000000	10192	19021	563468.912743

**Table 4** Significance of some cliques obtained using MR-CoC

Sub-network	Functional coherence > 70%		
	Biological function	Molecular process	Cellular component
	DNA binding, DNA-directed DNA polymerase activity, damaged DNA binding	Single-organism biosynthetic process, single-organism metabolic process, cellular biosynthetic process, organic substance biosynthetic process	Nucleoplasm, DNA polymerase complex
	Transferase activity, heterocyclic compound binding, organic cyclic compound binding, DNA binding	Regulation of primary metabolic process, regulation of cellular metabolic process, cellular macromolecule metabolic process, primary metabolic process	Intracellular organelle part, protein complex, DNA-directed RNA polymerase
	Signal transducer activity, transcription factor activity-binding, transcription regulatory region DNA binding	Regulation of cellular metabolic process, regulation of macromolecule metabolic process, nucleic acid metabolic process, regulation of gene expression, etc.	Nucleus
	Structural molecule activity, extracellular matrix structural constituent	Single-organism catabolic process, extracellular matrix organization, extracellular matrix disassembly, collagen catabolic process	Intracellular organelle lumen, extracellular region part, extracellular matrix
	G-protein coupled receptor activity, olfactory receptor activity	Response to a stimulus, cellular response to a stimulus, response to chemical, cell communication, G-protein-coupled receptor signaling pathway, sensory perception of smell	Plasma membrane, cell periphery

(continued)

**Table 4** (continued)

Sub-network	Functional coherence > 70%		
	Biological function	Molecular process	Cellular component
	Hydrogen ion transmembrane transporter activity, oxidoreductase activity, ubiquinol-cytochrome-c reductase activity	Hydrogen ion transmembrane transport, respiratory electron transport chain, cellular respiration, mitochondrial ATP synthesis coupled electron transport, mitochondrial electron transport, ubiquinol to cytochrome c	Organelle membrane, membrane protein complex, organelle envelope, respiratory chain
	Extracellular matrix structural constituent	Single-multicellular organism process, multicellular organismal development, system development, single-organism catabolic process	Intracellular organelle, membrane-bounded organelle, protein complex, endomembrane
	Ion binding, catalytic activity, hydrolase activity, ATP Binding, structure-specific DNA binding,	Response to stress, cellular response to a stimulus, cell cycle, DNA recombination, DNA repair, mismatch repair, meiotic cell cycle	Nuclear part, nuclear lumen, nucleoplasm, chromosome
	Tetrapyrrole binding, catalytic activity, metal ion binding, oxidoreductase activity	The single-organism metabolic process, small molecule metabolic process, catabolic process, oxidation-reduction process, lipid metabolic process	The bounding membrane of the organelle, endomembrane system, endoplasmic reticulum
	Cytokine receptor binding, receptor binding, growth hormone receptor binding, growth factor activity	Cytokine-mediated signaling pathway, positive regulation of tyrosine phosphorylation of Stat3 protein, regulation of JAK-STAT cascade, cellular response to cytokine stimulus	Interleukin-6 receptor complex, ciliary neurotrophic factor receptor complex

The cohesiveness of the sub-network is clearly showcased by the score measure. The computational time taken for different seed setup is given in the last column of Table 3. These executions are carried out in a system with Intel I7 processor and 8 GB RAM. The biological significances of the obtained cliques and non-cliques can be further studied to know their functionality. Some of the cliques and non-cliques along with their biological function, molecular process, and cellular component were extracted using the string database [18] query services listed in Table 4. The biological significances of the sub-networks are further studied to know their functionalities. The proposed MR-CoC can be applied for specific disease proteins to learn their functional tactics on the host proteins. They are used to assist the drug discovery, drug target identification, etc. The proposed methodology is useful in a distributed environment to mine the cohesive sub-network from the complex interaction networks. The computational time can be reduced considerably if the computation is carried out in a distributed setup. It will help to overcome the issues of big data in protein interaction networks.

This MR-CoC model can be used as one of the big data mining techniques for any kind of network dataset like social interaction networks, traffic analysis, and function complexity analysis in software engineering. This model can be applied to biclustering binary dataset in any domain.

## 6 Conclusion

Network module or subgraph mining is one of the emerging research areas. The proposed methodology score-based co-clustering algorithm with MapReduce model is attempted to mine the sub-networks from large networks like PIN. The objective is to parallelize the sub-network mining process within the commodity system using MR-CoC and to reduce the time constraints. It is devised to support the computational specialists to undergo these kinds of computational works in a distributed environment. Further, the same methodology can be carried out on protein interactions of different organisms, social network interactions, mobile network interactions, etc. The different methodologies can be incorporated into initial seed selection. This methodology can be optimized using meta-heuristic algorithms to mine optimal cohesive sub-networks.

## References

1. Structures of Life (2007)
2. Gowri, R., Rathipriya, R.: A study on clustering the protein interaction networks using bio-inspired optimization. *IJCII* **3**, 89–95 (2013)
3. Ekanayake, J., Pallickara, S., Fox, G.: MapReduce for data intensive scientific analyses. In: *Proceeding ESCIENCE '08 Proceedings of the 2008 Fourth IEEE International Conference on eScience*, pp. 277–284 (2008)



4. Chen, S., Schlosser, S.W.: Map-reduce meets wider varieties of applications. Intel Research Pittsburgh, Technical Report 2008, IRP-TR-08-05
5. Rosen, J., Polyzotis, N., Borkar, V., Bu, Y., Carey, M.J., Weimer, M., Condie, T., Ramakrishnan, R.: Iterative mapreduce for large scale machine learning
6. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. *Commun ACM* **51**, 107–113 (2008)
7. Aridhi, S., D’Orazio, L., Maddouri, M., Mephu, E.: A Novel MapReduce-Based Approach for Distributed Frequent Subgraph Mining. RFIA (2014)
8. Hill, S., Srichandan, B., Sunderraman, R.: An iterative MapReduce approach to frequent subgraph mining in biological datasets. In: ACM-BCB’12, pp. 7–10 (2012)
9. Gowri, R., Rathipriya, R.: Extraction of protein sequence motif information using PSO K-Means. *J. Netw. Inf. Secur.* (2014)
10. Gowri, R., Sivabalan, S., Rathipriya, R.: Biclustering using venus flytrap optimization algorithm. In: Proceedings of International Conference on Computational Intelligence in Data Mining CIDM, Advances in Intelligent Systems and Computing series, vol. 410, pp. 199–207 (2015)
11. Gowri, R., Rathipriya, R.: Protein motif comparator using PSO k-means. *Int. J. Appl. Metaheuristic Comput. (IJAMC)* **7** (2016)
12. Bader, G.D., Hogue, C.W.V.: An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinform.* **4** (2003)
13. Laix, L., Qinx, L., Linx, X., Chang, L.: Scalable subgraph enumeration in MapReduce. In: Proceedings of the VLDB Endowment, vol. 8, pp. 974–985
14. Pereira-Leal, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. In: *PROTEINS: Struct. Funct. Bioinform.* **54**, 49–57 (2004)
15. Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Müller, T.: Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *ISMB* **24**, 223–231 (2008)
16. Pinkert, S., Schultz, J., Reichardt, J.: Protein interaction networks—more than mere modules. *PLoS Comput. Biol.* **6** (2010)
17. Zhang, Y., Zeng, E., Li, T., Narasimhan, G.: Weighted Consensus Clustering for Identifying Functional Modules In Protein-Protein Interaction Networks
18. A Faster Algorithm for Detecting Motifs. In: 5th WABI-05, vol. 3692, pp. 165–177. Springer (2005)
19. Kashtan, N., Itzkovitz, S., Milo, R., Alon, U.: Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics* **20** (11), 1746–1758 (2004)
20. Gursoy, A., Keskin, O., Nussinov, R.: Topological properties of protein interaction networks from a structural perspective. *Biochem. Soc. Trans.* 1398–1403 (2008)
21. Schaeffer, S.E.: Graph clustering. *Comput. Sci. Rev.* 27–64 (2007)
22. Diestel, R.: *Graph Theory*. Springer (2016)
23. Ray, S.S.: Subgraphs, paths and connected graphs. In: *Graph Theory with Algorithms and its Applications*, pp. 11–24 (2013)
24. Bapat, R.B.: *Graphs and Matrices*. Springer, Hindustan Book Agency (2010)
25. Ke, H., Li, P., Guo, S., Guo, M.: On traffic-aware partition and aggregation in MapReduce for Big Data applications. *IEEE Trans. Parallel Distrib. Syst.* (2015)
26. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K.P., Kuhn, M., Bork, P., Jensen, L.J., von Mering, C.: STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucl. Acids Res.* **43**, 447–452 (2015)

# Design and Development of Hybridized DBSCAN-NN Approach for Location Prediction to Place Water Treatment Plant



Mousi Perumal and Bhuvaneswari Velumani

**Abstract** Water plays an important part in all living organisms on earth by balancing the entire ecosystem; the natural resource is being exploited and contaminated due to technological growth, urbanization, and human activities. The natural resource has now changed into a precious commodity, with which many businesses flourish. The objective of the work is to identify locations to set water purification plant near water bodies such as river, pond, and lake to reuse the contaminated water for agriculture and for other basic need using Spatial Data Mining (SDM). SDM operates on location-specific analysis stored in geo-databases which are a collection of spatial and non-spatial data. The spatial and non-spatial data for Coimbatore region is collected, and the location prediction for setting water treatment plant is done through DBSCAN-NN algorithm using SDM tools.

**Keywords** Spatial Data Mining · DBSCAN-NN · Geo-databases  
Location prediction

## 1 Introduction

India's hike in growing population has forced a brutal strain on all the natural resources of the country. Water, a natural resource, plays a major role in balancing the entire ecosystem. In India due to technological growth, urbanization, and human activities, the water resources are more exploited and contaminated. It is estimated that 80% of India's surface water is polluted and let into the rivers for more than 5 decades [1]. The runoff has been made an inconsistency in supply of safe water to all the people of India, which directly affects the ecology and indirectly affects the fertility of the humans. Most of the rivers in India are polluted, and the number has

---

M. Perumal (✉) · B. Velumani  
Department of Computer Applications, Bharathiar University, Coimbatore, India  
e-mail: Mousiperumal@gmail.com

B. Velumani  
e-mail: Bhuvanes\_v@yahoo.co.in

doubled in last 5 years [2, 3]. The natural resource has now changed into a precious commodity, with which many businesses flourish. In India, 12.4% of the people lack access to clean and safe drinking water [4]. Water contamination has led to many health and environmental issues, which has become a serious concern. In India, 7.5% of the total death annually is caused due to waterborne diseases. Water contamination is caused mainly due to the letting of untreated domestic sewage into the rivers [5]. In order to provide safe and clean water to the citizens, India needs an infrastructure development to treat the contaminated water. The infrastructure development could be made through Spatial Data Mining; a process used in discovering interesting patterns from large geo-databases [6]. Voluminous geographic data is continued to be collected with modern data acquisition techniques such as global positioning systems (GPS), high-resolution remote sensing, location-aware services and surveys, and Internet-based volunteered geographic information [7, 8].

The process of discovering hidden patterns and unknown information from large spatial data sets is called as Spatial Data Mining. The spatial option is designed to make spatial data management easier and more natural to users and applications of Geographic Information System (GIS) [9, 10]. The spatial data sets are collection of spatial and non-spatial attributes. The spatial attributes represent the information with respect to earth's location based on latitudinal and longitudinal position. Mining of spatial data set is found to be complex as the spatial relations are not represented explicitly in geo-databases [11–13]. Analysis of spatial data has become important for analyzing information with respect to location. So, analysis of spatial data requires mapping of spatial attributes with non-spatial attributes for effective decision making. Spatial data is also known as geospatial data which contains information about a physical object that can be represented by numerical values in a geographic coordinate system. Spatial Data Mining combines various tasks and different methods (computational, statistical, and visual) for effective analysis of data. Location prediction through Spatial Data Mining techniques (clustering, classification, association, etc.) plays an important role in many infrastructure settings such as dam construction, anaerobic digestion system, power grid plant, roadways planning [14–16]. The work concentrates on clustering of the spatial data to find appropriate locations for setting water treatment plant. The clustering of the data is a difficult task in the existence of physical constraints; a new approach is designed to find the relations among different spatial variables using geometric relations [11, 17, 18]. A field development model is developed based on spatial clustering; to provide solutions to all kinds of problems [12].

The objective of this paper is to develop a methodology to identify an optimal location to place water treatment plant near water bodies. The paper is organized as follows: Sect. 2 gives a detailed description of the methodology developed for location prediction. Section 3 describes the results and the discussions, followed by conclusion in Sect. 4.

## 2 Framework for Location Prediction Analysis

The framework is proposed to identify an optimum location to place a water treatment plant in Noyyal River basin of Coimbatore district. The objective is achieved in three phases; as a first phase, the collected data is preprocessed and the non-spatial data is converted to spatial data. The spatial data model is designed to cluster spatial points using hybridized DBSCAN-NN algorithm for location prediction. Finally, an optimal location is predicted to place a water-purifying plant on Noyyal River basin based on the exploratory data analysis (Fig. 1).

### Data Selection

The data for the proposed work is collected by various means of information available (Web sites and corporation office) for Coimbatore district. The non-spatial attributes such as name, address of schools, colleges, hospitals, parks, lakes that are present in the Coimbatore region surrounding the Noyyal River basin for distance of about 115 km. For easy administration, the study area is divided into five zones (north, south, east, west, and central). The data set collected consists of 900 instances which had some missing values; fieldwork has been carried out to collect such details that were not available. Vector data of Coimbatore district with scale ranging between 1:10,000 and 1:100,000 is used as a base map for the analysis. A related spatial analysis using the map created could be done for different variables that are interrelated. The variable selection was made using the interrelation between the variables; the variables showing high communality for almost all the variables are selected.

### Data Preprocessing and Transformation

The data collected is to be preprocessed and transformed from non-spatial to spatial data. A total of 900 instances collected consist of missing values, duplicate records, unwanted extra information, and irrelevant data. Deletion of irrelevant data and permanently closed schools, colleges, hospitals are done using the domain experts from the Coimbatore Corporation Office. The normalization of the data is done to avoid the redundancy in the data collected. Each data entry is assigned with the unique identifier (Z\_id) that maps the corresponding zone details to the data.

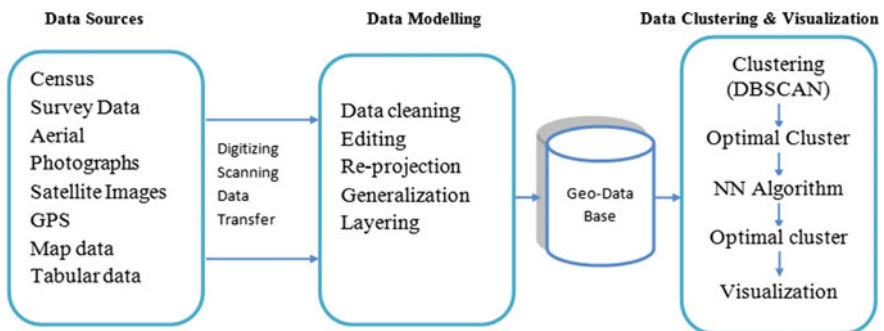


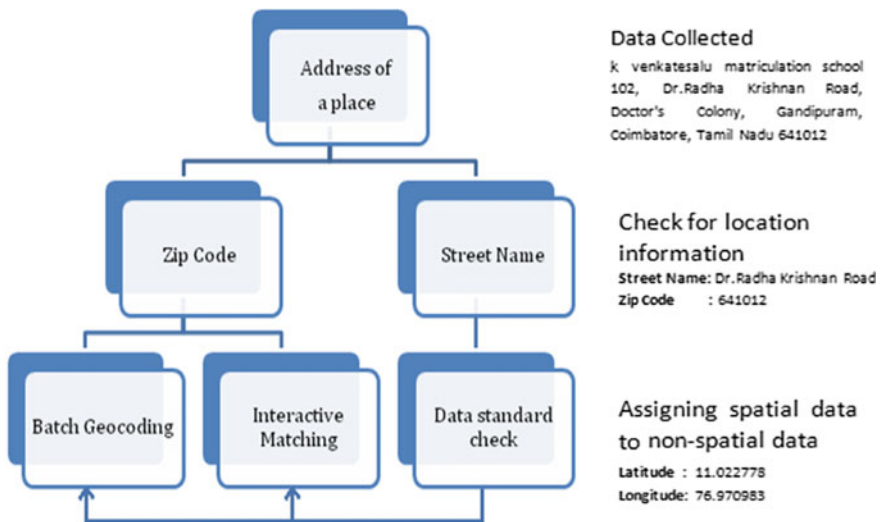
Fig. 1 Framework for location prediction analysis

The preprocessed (non-spatial) data is to be converted to spatial data for location-specific analysis. Geo-coding is a process by which it assigns latitude (X) and longitude (Y) directly to the non-spatial attributes in the data file. The values for latitude and longitude are based on the World Geodetic System of 1984 (WGS84). WGS84 format is easy to merge with other data for further analysis. Figure 2 shows a step-by-step process of data standardization and geo-coding process with an example.

**Spatial Data Modeling**

The data schema for representing the data set is done using Oracle 11g. The data schema is designed using the function SDO\_GEOMETRY to represent the geospatial attribute and spatial relations. Separate tables are created for each zone with the geometric features (point, line, and polygon). Table 1 represents the geometric features used for the work, and Table 2 represents the schema design.

Spatial indexing (binary tree-based indexing) is done to support spatial selection and to retrieve geometric relationships from the data set. Metadata is created for geometric relations in order to index and use geometric functions.



**Fig. 2** Geo-coding process

**Table 1** Geometric feature of the proposed work

Point	Represents the latitude and longitudinal position of schools, colleges, hospitals, parks, lakes
Line	Represents the Noyyal River pathway in the district
Polygon	Represents the administrative zone boundaries of the Coimbatore district

**Table 2** Schema model

Column name	Description	Data type	Constraint
C_Id	Unique college identification number	Number	Primary key
C_Name	College name	Varchar2(32)	–
C_Shape	Geometric feature(point)	Public. SDO_GEOMETRY	Point
Z_Id	Zone identification number	Number	Foreign key

### Spatial Clustering

The clustering of the data is done to find the densely clustered region of the Coimbatore region for location prediction to place water treatment plant in various zones. The clustering of the data is done using hybridized DBSCAN-NN approach. As a first phase, DBSCAN algorithm is used to cluster the data by varying the distance (5–25 km) and epsilon value (0.1–1); the distance and epsilon for grouping clusters are varied until number of clusters generated are same [19, 20]. The cluster that groups the maximum number of neighbors is opted as optimal cluster. In the second phase, the optimal cluster retrieved from DBSCAN is iterated using NN algorithm. The cluster that retrieves the maximum number of park is treated as optimal cluster and that location is predicted to set a water treatment plant [21, 22]. The water treatment is proposed to be set near the park to make use of the recycled water before it drains into the Noyyal River.

### Visual Analysis

The cluster points are visualized to verify the accuracy of the clusters generated. The visual verification is carried out using ArcGIS. The shapefile Tamil Nadu is created as the base map. The Coimbatore district boundary layer is created using the latitude and longitude boundaries of the district. The Noyyal line string is plotted over the Coimbatore region. The projection coordinates are correctly matched to obtain accurate results. The optimal cluster points are post-processed, and the positions are plotted over the Coimbatore region.

## 3 Results and Discussion

A total of about 900 schools, colleges, hospitals, and dispensaries are collected for the study out of which 850 were used for the further analysis. Permanently closed schools and colleges are removed. The duplicated data is also deleted using the preprocessing techniques. The total number of schools, hospitals, colleges of Coimbatore district is listed out in Table 3.

**Table 3** Preprocessed data set

Category	Collected data	Data after preprocessing
Schools (middle, high, higher secondary)	450	442
Colleges	120	112
Hospitals and dispensaries	300	296

Based on the results obtained from Perumal et al., [23] the optimal clustered data is fed into NN algorithm to find the optimal clusters with respect to park by varying distance and epsilon for respective zones. The distance (5–25 km) and epsilon value (0.1–1) for grouping clusters are varied until same numbers of clusters are generated.

The cluster chosen for location prediction is based on the park location, which groups more number of neighbors to construct water treatment plant and use the water in the parks before it drains into the Noyyal River.

Table 4 presents the clusters generated for each zone with different distance and epsilon values. It is inferred that one cluster groups large number of parks for each zone and that cluster is opted as optimal cluster. Table 5 presents the total number of optimal clusters generated zone-wise with respect to park. The optimal clusters are the clusters where the location has been predicted to place water treatment plant in each zones.

Figures 3 and 4 give the cluster plot of the optimal clusters of the two zones: north and east. Table 6 presents the location predicted for water treatment plant zone-wise.

Based on the experimental run, it is found that Koundampalayam, Sundrapuram, Peelamedu, Vadavalli, and Tatabad are considered as optimal location for the constructing water treatment plant. The cluster groups maximum number of instances and parks. The outcome of the work contributes toward the surface water quality maintenance. It is estimated that an average of 4 million liters of water would be processed daily for watering the parks and 50 lakh hectares of agricultural land.

### Visual Analysis

The map is created to visualize the cluster output using ArcGIS. The cluster points' latitude and longitude are layered on the Coimbatore region map. Figures 5, and 6 represent map for visualization of the location predicted with respect to cluster zone-wise. The map visualization of optimal location shown in the icon of the park is mentioned in the zoomed area of the map. From the map, it is inferred that cluster groups are generated above the Noyyal River basin for all zones, and no clusters were generated below the Noyyal River basin since the lower riverbed is sparsely populated than the upper.

**Table 4** Clusters generated for various distance and epsilon values for various zones with respect to park

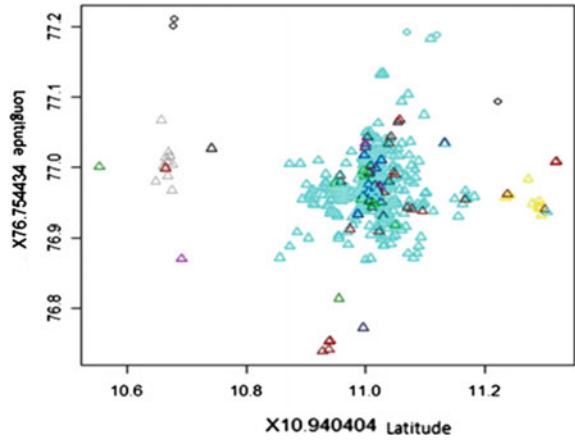
Zone no.	Distance (km)	Epsilon value (neighborhood value)								
		0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18
1	5	5	4	4	3	2	1	1	1	1
	10	11	9	4	3	14	3	2	1	1
	15	15	11	4	2	2	2	1	1	1
	20	17	11	5	3	3	2	2	1	1
	25	19	12	7	5	3	2	2	1	1
2	5	6	6	6	6	5	3	3	1	1
	10	15	8	7	5	4	3	3	3	3
	15	18	10	7	7	4	3	2	2	1
	20	21	12	9	7	5	4	3	1	1
	25	20	14	10	9	6	5	3	3	1
3	5	10	7	7	5	5	3	3	3	1
	10	11	10	9	9	7	5	3	3	1
	15	21	11	10	8	5	3	2	1	1
	20	21	11	11	9	9	5	2	1	1
	25	21	10	10	9	9	7	7	3	3
4	5	15	6	4	3	1	1	1	1	1
	10	17	8	7	5	3	2	1	1	1
	15	18	10	8	4	1	1	1	1	1
	20	21	16	15	9	4	3	3	2	1
	25	20	11	9	6	3	3	1	1	1
5	5	5	3	2	1	1	1	1	1	1
	10	8	4	4	2	1	1	1	1	1
	15	11	6	5	4	3	3	3	3	1
	20	11	5	3	3	3	3	3	3	1
	25	11	7	6	4	4	3	3	1	1

**Table 5** Total number of optimal clusters zone-wise

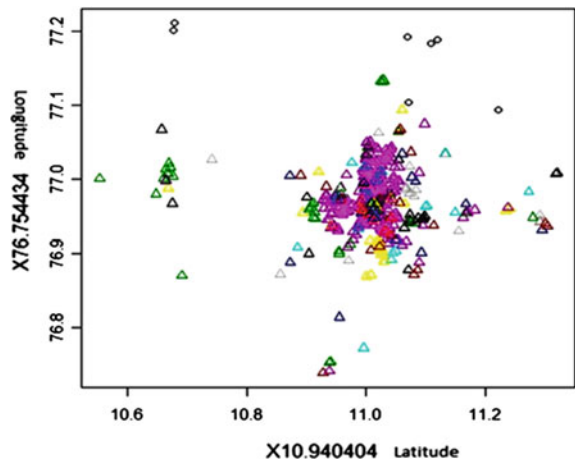
Zone no	Distance (km)	Epsilon value	Cluster no	Total number of			
				Colleges	Hospitals	Parks	Schools
1	15	0.2	1	1	12	3	41
2	20	0.08	1	8	38	14	64
3	5	0.04	1	14	65	4	66
4	10	0.08	1	1	57	12	52
5	15	0.06	1	1	82	12	63



**Fig. 3** Optimal clusters view with respect to 15 km and epsilon value 0.2 of zone 1 (north)



**Fig. 4** Optimal clusters view with respect to 5 km and epsilon value 0.04 of zone 3 (east)



**Table 6** Location predicted for water treatment plant zone-wise

Zone	Location	Parks	No. of neighbors
1-North	Koundampalayam	Koundampalayam Children’s Park	57
2-South	Sundrapuram	Kuruchi Park, Children’s Park	124
3-East	Peelamedu	Peelamedu Children’s Park, Kaveri Park	149
4-West	Vadavalli	TNAU Botanical Garden, Vadavalli Panchayat Park	122
5-Central	Tatabad	Government Park-Ram Nagar, Aringar Anna Park	158

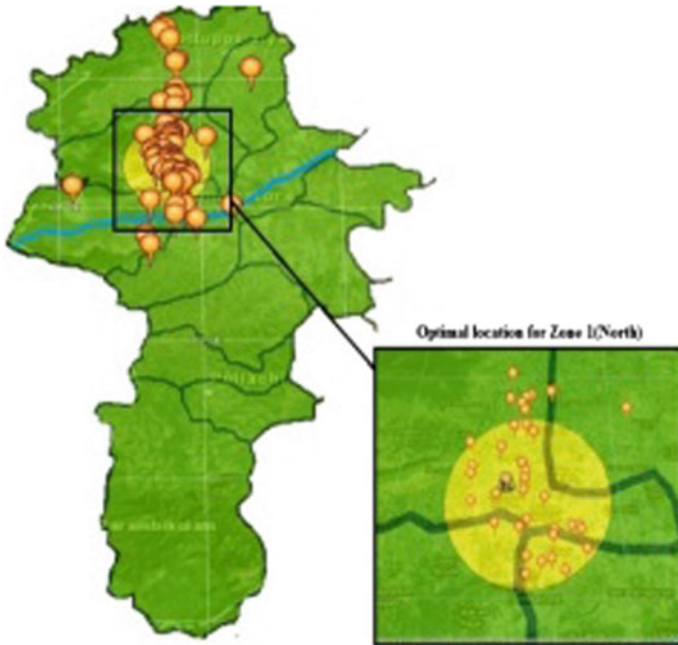


Fig. 5 Map view of north zone



Fig. 6 Map view of south zone

## 4 Conclusion

The methodology helps in location prediction to place water treatment plant in Noyyal River basin. The experimental study identifies optimal location to construct water-purifying plant on Noyyal River basin. The optimal clusters are retrieved for four zones of Coimbatore region. The best locations are opted with nearest watering places (parks and agricultural land) to reuse the processed water from water treatment plant. The exact location for construction of water treatment plant is not given as other constraints such as budget roadways elevation are not taken into consideration for this work.

## References

1. Du, Y., Liang, F., Sun, Y.: Integrating spatial relations into case-based reasoning to solve geographic problems. *Known.-Based Syst.* (2012)
2. Lee, A.J.T., Hong, R.W., Ko, W.M., Tsao, W.K., Lin, H.H.: Mining frequent trajectory patterns in spatial-temporal databases. *J. Inf. Sci.* **179**, 2218–2231 (2009)
3. Wang, S., Ding, G., Zhong, M.: *Big Spatial Data Mining*, pp. 13–21. IEEE (2013)
4. Bai, H., Ge, Y., Wang, J., Li, D., Liao, Y., Zheng, X.: A method for extracting rules from spatial data based on rough fuzzy Sets. *J. Knowl. Based Syst.* **57**, 28–40 (2014)
5. Bi, S., et al.: *Spatial Data Mining in Settlement Archaeological Databases Based on Vector Features*, pp. 277–281. IEEE (2008)
6. De Moraes, A.F., Bastos, L.C.: Pattern Recognition with Spatial Data Mining in Web: An Infrastructure to Engineering of the Urban Cadaster, pp. 1331–1335. IEEE (2011)
7. Brimicombe, A.J.: A dual approach to cluster discovery in point event data sets. *Comput. Environ. Urban Syst.* **31**(1), 4–18 (2007)
8. Spielman, S.E., Thill, J.C.: Social area analysis, data mining and GIS. *Comput. Environ. Urban Syst.* **32**(2), 110–122 (2008)
9. He, B., Fang, T., Guo, D.: Uncertainty in Spatial Data Mining, pp. 1152–1156. IEEE (2004)
10. Shekhar, S., Zhang, P., Huang, Y., Vatsavai, R.R.: Trends in spatial data mining. In: Kargupta, H., Joshi, A. (eds.) *Data Mining: Next Generation Challenges and Future Directions*. AAAI/MIT (2003)
11. Du, Qin, Q., Wang, Q., Ma, H.: Reasoning about topological relations between regions with broad boundaries. *Int. J. Approx. Reason.* **47**, 219–232 (2008). [10.1016/j.ijar.2007.05.002](https://doi.org/10.1016/j.ijar.2007.05.002)
12. Yasmin, M.: *Dynamic Referencing Rules Creation Using Intelligent Agent in Geo Spatial Data Mining*. IEEE (2012)
13. Shengwu, H.: *Method Development about Spatial Data Mining and Its Problem Analysis*, vol. 2, pp. 144–147. IEEE (2011)
14. Shekhar, S., Lu, C.T., Zhang, P.: A unified approach to detection spatial outliers. *GeoInformatica* **7**, 139–166 (2003)
15. Peng, S., Fang, J., Han, C., Cheng, Z.: VegaMinerPOI: A Spatial Data Mining System for POI Datasets, pp. 1–4
16. Lee, A.J.T., Hong, R.W., Ko, W.M., Tsao, W.K., Lin, H.H.: Mining spatial association rules in image databases. *Inf. Sci.* **177**, 1593–1608 (2007)
17. Xiao ping, L., Zheng yuan, M., Jian hua, L.: A spatial clustering method by means of field model to organize data. In: *Second WRI Global Congress on Intelligent Systems (GCIS)*, pp. 129–131 (2010)

18. Wang, Z., et al.: Cluster Analysis Based on Spatial Feature Selecting in Spatial Data Mining, pp. 386–389. IEEE (2008)
19. Anselin, L., Schockaert, S., Smart, P.D., Twaroch, F.A.: Generating approximate region boundaries from heterogeneous spatial information: an evolutionary approach. *J. Inf. Sci.* **181** (2), 257–283 (2011)
20. Shi, W.Z., Tong, X.H., Liu, D.J.: A least squares based method for adjusting the boundaries for area objects. *Photogramm. Eng. Remote Sens.* **71**(2), 189–195 (2005)
21. Wang, S, Yuan, H.: Spatial Data Mining in the Context of Big Data, pp. 486–492. IEEE (2013)
22. Zaiane, O.R., Lee, C.-H.: Clustering spatial data in the presence of obstacles: a density-based approach. In: Proceedings. International Database Engineering and Applications Symposium, pp. 214–223 (2002)
23. Perumal, M., Velumani, B.: Framework to find optimal cluster to place water purification plant using spatial clustering-DBSCAN. In: Materials Today Proceedings, vol. 25 (2016)

# Coupling on Method Call Metric—A Cognitive Approach



K. R. Martin, E. Kirubakaran and E. George Dharma Prakash Raj

**Abstract** Software development is a greatly multifaceted and intellect-oriented movement. In the early days of program development, developers engraved programs using machine language in which programmers consumed added time, thinking about a particular machine's instruction set than the difficulty at hand. Progressively, programmers wandered to higher level languages. In order to progress software using higher level languages, there are different approaches and selection of software development approach depending on the type of application to be established. Aspect-oriented software development (AOSD) is a novel paradigm in software development that addresses certain concerns in software development that regards modularization as an important aspect. The practices of AOSD create the possibilities to modularize, crosscutting the concerns of a system. Analogous to objects in object-oriented software development, aspects in AOSD may ascend at any point of time of the software life cycle, containing requirements specification, design, implementation, etc. As this effort is built on empirical validating aspect-oriented metrics, coupling on method call (CMC) metric is selected. This work focuses on empirical validation of the metrics. Novel metric is cognitive weighted metric, which is calculated for CMC to calculate the coupling complexity value of the aspect.

**Keywords** Software metrics • Aspect-oriented software development (AOSD) • Aspect-oriented programming (AOP) • Coupling on method call (CMC) • Cognitive weighted coupling on method call (CWCWC)

---

K. R. Martin (✉)  
St. Joseph's College, Tiruchirappalli, India  
e-mail: krmartincs@gmail.com

E. Kirubakaran  
BHEL, Tiruchirappalli, Tamil Nadu, India  
e-mail: ekirubakaran@gmail.com

E. George Dharma Prakash Raj  
Bharathidasan University, Tiruchirappalli, India  
e-mail: georgeprakashraj@yahoo.com

## 1 Introduction

Aspect-oriented programming (AOP) spreads the traditional object-oriented programming (OOP) model to expand code reuse across different object hierarchies. AOP can be used with object-oriented programming [1]. AspectJ is an implementation of aspect-oriented programming for Java. AspectJ adds to Java just one new concept, a join point and that improves name for an existing Java concept. It adds to Java only rare novel paradigms: point cuts, advice, inter-type declarations, and aspects [2]. Cognitive complexity measures characterize the human effort needed to accomplish a chore or struggle in comprehending the software code.

Coupling is an internal trait of the software which is a measure of the degree of system interdependence of the components of the software. Coupling is considered to be one of the necessary goals in software construction, which will eventually lead to better maintainable, reusable, and reliable software products [3]. Various coupling measures have been proposed for aspect-oriented software. Aspect-oriented software development is not an alternate for object-oriented paradigm, but aspect orientation complements object orientation. So many of the metrics are used in the aspect-oriented systems may be extended from object-oriented software.

AspectJ has no **CWCMC** metric to quantify all the inheritances of an OO system recommended by earlier researchers. So, the necessity arises for the cognitive weighted coupling on method call (CWCMC) for the aspect-level inheritance measurement. Hence, the significant objective is to bring out a CWCMC metric to the complexity of different types of inheritance.

## 2 Literature Review

There exist several metrics that were established and published by researchers for OO systems. The metric suite of Aloysius and Arockia Sahaya Sheela [4] is regarded as benchmark of OO metrics. The CK metric suite consists of weighted method per class (WMC), depth of inheritance tree (DIT), response for a class (RFC), number of children (NOC), lack of cohesion of methods (LOCM), and coupling between objects (CBO).

Kulesza [5] introduced many aspect-oriented metrics which included aspect-oriented coupling metrics as well. The metrics that the study used were the extension of the metrics suite from object-oriented metrics. Moreover, this work collected the value for the metrics from the software using the tool developed for this purpose.

Gradecki and Lesiecki [6] mentioned about aspect-oriented metrics in their work that reports the development of several metrics for AOP design paradigm. They also study the implementation of the metrics and throw light on the directions for further work.

Bartsch and Harrison [7] presented a quantitative learning that assesses the helpful and unsafe effects of AOP on maintenance and happenings of a Web information system. The learning also considered the pros and cons of AOP on coupling measures when compared to the object-oriented implementation of the same.

The AspectJ Team [2] studied aspect-oriented coupling and cohesion measures for aspect-oriented systems. This learning is planned to frame an idea about the coupling, cohesion measures, and framework all along with tool support for the coupling measures.

Kiczales et al. [1] analyzed improving the design of cohesion and coupling metrics for aspect-oriented software development. This study focuses on developing metrics for better calculation of coupling and cohesion values.

### 3 Methodology

#### 3.1 Existing Metric

The count of the modules or interfaces stating methods that are called by a given module is the coupling on method call (CMC). This metric is analogous to the OO metric—coupling between objects (CBO). This metric is divided into two different metrics, called coupling on method call (CMC) and coupling on field access (CFA). These two metrics separate coupling on operations from coupling on attributes. Aspect introductions must be taken into account when the feasibly appealed methods are determined.

#### 3.2 Proposed Metric

**Cognitive Weighted Coupling on Method Call (CWCMC):** Many metrics were developed by the pioneers in the field for AOP systems. One of the metrics proposed by Ceccato and Tonella [3] is CMC. CMC sums up the number of modules or interfaces stating methods that are invoked by a specific module. This particular metric does not measure all possible return types. The proposed metric, CWCMC, counts the modules or interfaces declaring different return type methods that are invoked by a module and multiplied by the number of parameters.

$$\text{That is, } CWCMC = (VO * WFVO + IN * WFIN + FL * WFFL + LO * WFLO + DO * WFDO) + NOP \quad (1)$$

Metrics are important key factors in every area such as software engineering, network environment, and cloud environment. Metrics play a vital role in each area differently. Cloud metrics are essential at the stage of deciding what cloud offering is best suited to meet the business and technical requirements. In addition, other parts of the cloud ecosystem can be influenced through the use of metrics like accounting, auditing, and security. In a cloud environment, metrics are proposed for efficient resource scheduling, data security, and also for maintenance.

## **4 Empirical Metric Data Collection and Criteria of Evaluation**

This part of the paper deliberates the CWCMC metric, empirical data, collection statistics, analysis, and its implication.

### **4.1 CMC Metric**

For empirical analysis, CMC metric is selected for AO software. This metric is used to find the complexity of various return type methods using cognitive approach.

### **4.2 Calibration**

An experiment is conducted to assign cognitive weights to many of the return types for methods. An intellectual capacity test has been carried out for a set of students to measure the time consumed to understand the complexity of aspect-oriented program with regard to various return types. The group of students nominated had appropriate acquaintance in evaluating the aspect- and object-oriented programs as they had studied courses in AspectJ programs. The student set consists of thirty from rural areas and another thirty from urban, and they were designated to be included in the intellectual capacity test.

The time used by students to understand the semantics of the programs was verified after the completion of each exercise. The time taken for comprehension of all these programs was recorded and the average time spent to comprehend was noted down. In every case, five exercises have been given to understand. In total, twenty-five different mean times were registered. Average time was computed for each and every program from the time consumed by the every student individually.

The normal understanding time, for programs which are developed based on aspect-oriented programming is listed. Also, the mean time is calculated for each group and tabulated.



**Table 1** Mean values of different return types

Program	Average comprehension time (h)			
	Integer	Float	Long	Double
Pgm_1	0.16	0.30	0.4	0.50
Pgm_2	0.17	0.28	0.4	0.50
Pgm_3	0.20	0.30	0.4	0.47
Pgm_4	0.20	0.30	0.4	0.50
Pgm_5	0.20	0.30	0.4	0.50
Mean	0.169	0.275	0.389	0.46
STD_DEV	0.314	0.335	0.299	0.574

## 5 Statistical Analyses

For each of the return types, the mean was designated as a degree of central tendency. Table 1 shows the statistical calculation of different return types.

A standard deviation near 0 indicates that the data points incline to be actual near the mean of the set.

## 6 CWCMC

Of the several metrics that have been proposed for AOP systems, one of them is given by Ceccato and Tonella [3] and is called CMC. This metric counts the modules or interfaces declaring methods that are possibly called by a specific module. This metric does not consider all of the return types. The proposed metric, called CWCMC, counts the modules or interfaces declaring different return type methods that are possibly invoked by a particular module is added by number of parameters.

$$\begin{aligned}
 \text{CWCMC} = & (\text{VO} * \text{WFVO} + \text{IN} * \text{WFIN} + \text{FL} * \text{WFFL} + \text{LO} * \text{WFLO} + \text{DO} * \text{WFDO}) \\
 & + \text{NOP}
 \end{aligned}
 \tag{2}$$

Here,

CWCMC is overall cognitive complexity of coupling on method call.

WFIN is weighting factor of integer return type.

WFVO is weighting factor of void return type.

WFFL is weighting factor of float return type.

WFDO is weighting factor of double return type.

WFLO is weighting factor of long return type.

NOP is number of parameter.

**Table 2** Weight value of each return type of method

Return type	Value of the weight
WFVO	0.1
WFIN	0.2
WFFL	0.3
WFLO	0.4
WFDO	0.5

**Table 3** Coupling complexity metric value for the example program

Program#	Existing metric value (CMC)	Proposed metric value (CWCMC)
1	2	0.2

Each of the weighting factors of each return type is shown in Table 2, which uses the method discussed in the empirical metric data collection. The weight value is calculated based on the mean value of different return type. To normalize the act in place of value to gain appropriate load value. The following table explained the rounded values of each return type that is called weighting factor of each return type. Here, void = 0.1 is a default weight value (Table 3).

## 7 Properties of Data Collection

The properties defined by Kulesza [5] were used for the data collection progression and are designated as given below.

**Accuracy:** The greater value shows that the difference between the actual data and measured data, and the lesser value shows the accuracy. The variance between CWCMC and CMC is lesser so the accuracy is higher.

**Replicability:** This means that the analysis can be completed at different times by different people using the same condition, and results are not significantly different. Data are taken from rural and urban PG students at different times.

**Correctness:** Data were collected corresponding to the metrics description. The value of CWCMC is collected and calculated through the CMC metric.

**Precision:** Data are conveyed by quantity of decimal places. Fewer decimal place displays a lower accuracy. The decimal place of the data is high (i.e. 0.466). So it shows a higher accuracy.

**Consistency:** It sums the differences with the metric values when collected using dissimilar tools by dissimilar people. Accordingly, it is found that the difference between the existing metric—CMC and proposed metric—CWCMC by giving different programs by different students.

The above properties are very common to be in every environment such as network and cloud. Cloud is now a crucial paradigm for outsourcing diverse computer needs of institutions. So, there is a need to propose metrics based on cloud with satisfying these properties.

## 8 Comparative Study

A comparative study made with the metric proposed by Ceccato and Tonella [3] is CMC. CMC sums the number of modules or interfaces proclaiming methods that are feasibly invoked by a known module. The current CWCMC metric is one phase ahead of existing CFA metric. This metric does not measure the several return types. The proposed metric, called CWCMC, counts amount of modules or interfaces asserting different return type methods that are possibly called by a known module is multiplied by quantity of parameter. One of the reward of CWCMC metric is that it proceeds with cognitive weights into consideration and data collection pleases the Kulesza [5] properties.

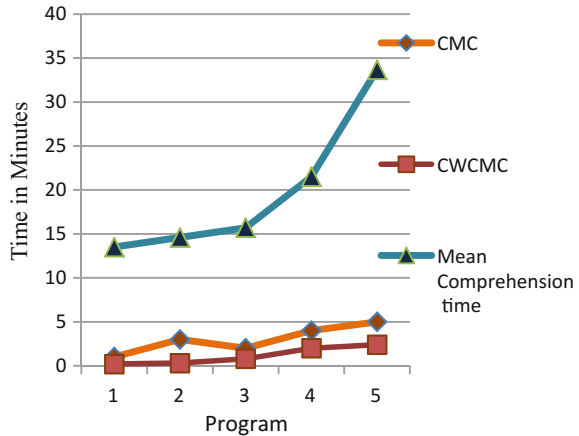
To associate the proposed metric, a comprehension test was implemented with rural and urban degree pupils. There were sixty pupils who participated in the test; the pupils were projected using five different programs in AspectJ for the intellectual capacity test. The test was conducted in order to find out the output of the given programs. The measure of time to complete the test is noted in minutes. The mean time taken by all the pupils was calculated. In the following Table 4, a comparison has been made with CMC, CWCMC, and the intellectual capacity test result.

CWCMC computes the quantity of modules or interfaces with different return type methods that are feasibly called by a given module is added by quantity of parameter. It is an improved pointer than the existing CMC. The mass of each return type is intended by using cognitive weights and weighting issue of return type related to which is recommended by Wang [8]. It originates that the resultant value of CWCMC is higher than the CMC. This is because, in CMC, the weight of each field is expected to be one. But, including cognitive weights for calculation of the CWCMC is more accurate because it reflects different return types. The results are shown in Table 4. A correlation analysis was done between CMC versus comprehension time with  $r = 0.873828$  and CWCMC versus comprehension time with  $r = 0.913323$ . CWCMC has more absolutely correlated than CMC (Fig. 1).

**Table 4** Complexity metric values and mean comprehension time

Prg#	Existing metric value (CMC)	Proposed metric value (CWCMC)	Mean comprehension time
1	1	0.2	13.5
2	3	0.3	14.6
3	2	0.8	15.7
4	4	2	21.5
5	5	2.4	33.67

**Fig. 1** Complexity metric values versus mean comprehension time



## 9 Conclusion and Future Scope

A CWCMC metric for determining the aspect-level complexity has been computed. CWCMC comprises the mental complication due to various return types and quantity of parameters. CWCMC has demonstrated that the complexity of the aspect receiving, which is grounded on the cognitive weights of the several return types of fields. The allocated cognitive weight of the several return types is authenticated using the intellectual capacity test. The metric is assessed through an experiment and verified to be an improved indicator of the aspect-level complexity. The metrics are constantly used in every environment. In future, more metrics can be applied in cloud environment also.

## References

1. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwin, J.: Aspect-Oriented Programming, pp. 220–242. Springer, Berlin, Heidelberg (1997)
2. The AspectJ Team: The AspectJ Programming Guide (2003)
3. Ceccato, M., Tonella, P.: Measuring the Effects of Software Aspectization: WARE (2004)
4. Aloysius, A., Arockia Sahaya Sheela, G.: Aspect oriented programming metrics—a survey. *Int. J. Emerg. Trends Comput. Commun. Technol.* **1**(3), 125–130 (2015)
5. Kulesza, U.: Quantifying the effects of aspect-oriented programming: a maintenance study. In: 22nd IEEE International Conference on Software Maintenance 2006, ICSM'06. IEEE (2006)
6. The AspectJ Team: The AspectJ Programming Guide (2001). Gradecki, J.D., Lesiecki, N.: Mastering AspectJ—Aspect-Oriented Programming in Java (2003)
7. Bartsch, M., Harrison, R.: An evaluation of coupling measures for AspectJ. In: LATE Workshop AOSD (2006)
8. Wang, Y.: On cognitive informatics. In: IEEE International Conference on Cognitive Informatics, pp. 69–74 (2002)

# An Intrusion Detection System Using Correlation, Prioritization and Clustering Techniques to Mitigate False Alerts



Andrew J. and G. Jasper W. Kathrine

**Abstract** Intrusion detection system (IDS) is one of the network security tools which monitors the network traffic for suspicious activity and alerts the network administrator. In large networks, huge volumes of false alerts are generated by IDS which reduces the effectiveness of the system and increases the work of the network administrator. The false incoming alerts raised by IDS lower the defence of network. In this paper, post-correlation methods such as prioritization and clustering are used to analyse intrusion alerts. The proposed framework uses prioritization to classify important and unimportant alerts and clustering approaches by correlating the alerts. Scalable distance-based clustering (SDC) is applied to further reduce the false alerts efficiently.

**Keywords** Intrusion • IDS • Correlation • Prioritization • Clustering  
SDC

## 1 Introduction

Nowadays, network attacks are growing, and security mechanisms are required to protect the network. Deploying security devices such as intrusion detection and prevention systems (IDPS), firewall, anti-malware helps the information technology organizations to protect their network from unwanted harmful traffic. An IDS plays a significant role in defending the network in IT systems. An IDS is aimed to monitor the computer network by detecting the malicious behaviour so that it can be reported to network analyst in the form of alerts. There are two common types of IDS which are used to detect attacks and access violations: signature-based and

---

Andrew J. (✉) · G. J. W. Kathrine  
Department of Computer Sciences Technology, Karunya University,  
Coimbatore, India  
e-mail: andrewj@karunya.edu

G. J. W. Kathrine  
e-mail: kathrine@karunya.edu

anomaly-based [1]. A signature-based detection technique stores the signatures of intrusion into the database. When the intrusion or attack occurs, the IDS checks whether the signature matches with the signatures that are stored in the database. If it matches, IDS generates an alert. An anomaly-based detection technique stores the network's normal behaviour into the database. When intrusion occurs which is different from the normal or past behaviour, then the alert is generated by IDS [2].

Though IDSs are effective in identifying attacks, it generates large number of false alerts. These alerts are examined by the network analyst in order to take actions against the malicious activities. But analysing this huge volume of alerts manually is difficult and time-consuming. According to [3], low-level and high-level alert operations are introduced to overcome this. Low-level alert management operations examine each alert individually. In high-level alert management, operations such as correlation, clustering and aggregation examine the set of alerts.

Alert correlation process has more advantages other than reducing huge volume of alerts [4]. False-positive alerts are reduced by correlation. This also helps to find the relationship between alerts from multiple IDSs. Creating a clear view of malicious events in networks and assistance to correct for timely decision-making against intrusions is another importance of correlation. Generally, an intruder attacks the network using multi-stage attack scenarios. IDS generates only low-level alerts for each step of an attack and cannot identify multi-stage attacks directly. But using alert correlation, this can be easily detected and the next step of the intruder can also be predicted.

Grouping up of similar objects into same category is known as clustering. In [5], density-based spatial clustering applications with noise (DBSCAN) algorithm is used to identify the clusters in large spatial data sets. This also efficiently classifies the data which are noise [5]. SDC does not require predefined number of clusters, and it also identifies noise.

This paper describes a method to analyse IDS alerts by finding the relationship between two alerts based on previous alert history. Clustering and prioritization are followed by correlating two alerts. Alert correlation techniques help to create high-level alert or meta-alert which significantly reduces the number of alerts to be verified by the analyst. To evaluate the processing of alerts based on its severity, alert prioritization is used. The priority levels are classified as high and low priorities. The unimportant alerts, i.e. the false alerts will be assigned as low priority. The alerts generated by one or more IDS are stored in a centralized database. These alerts are always mixed with irrelevant and duplicate alerts. This will increase the effort of the analyst while identifying successful alerts. Hence, alert clustering is used to group the alerts of same kind into single cluster.

The paper is organized as follows: Section 2 discusses related works; Section 3 describes our proposed system which minimizes the alerts; Section 4 presents the identified alert evaluation metrics; Section 5 presents the experimental results; finally, Sect. 6 concludes the paper.

## 2 Related Works

In the recent years, many approaches have been proposed for alert minimization in IDS. This section makes a description of different techniques proposed to reduce intrusion alerts based on correlation, prioritization and clustering.

Alert correlation process includes alert normalization, preprocessing, fusion, verification, attack session reconstruction and attack focus recognition [6]. The experimental result shows that effectiveness of each component depends on the data set that is being analysed. Rather than correlating alerts from single IDS, researchers have also proposed architectures which correlate alerts from multiple IDS. An architecture has been designed for collaborative IDSs which is used to analyse and integrate the alert quality generated by multiple IDSs [7]. To efficiently work with alerts generated from multiple IDSs, the proposed system uses alert correlation which identifies the occurrence of specific alerts and eliminates false alerts. In [8], an IDS system is proposed which detects the attacks like distributed denial-of-service (DDoS) using aggregator and correlator. The system is able to reduce 5.5% of repeated alerts from the original volume of data. The correlator used here also constructs attack scenarios for multi-stage attacks and detects large-scale attacks in real time.

In [9] describes an approach to reduce the number of IDS sensors for critical assets, using attack graph. The priority is assigned to alarms based on the distance to critical assets. In this paper, predictive nature of the attack graph helps to achieve predictive context of the attack response. Alsubhi et al. [10] proposed a fuzzy-logic-based scoring and prioritization technique for alerts generated by Snort. The results of the alert scoring show that with a grouping function, the alerts are reduced by 64.83% and without grouping function it reduces 92.57%.

An alert clustering approach with data mining and machine learning techniques to reduce the number of false-positive alerts in IDS has been proposed in [11]. Three clustering-based algorithms for signature-based IDS are explained in [12]; they are threshold-based clustering, threshold-based clustering with merging and adaptive clustering which limits the number of signatures being compared. The experiments are performed on KDD-99 intrusion detection data set. A graph clustering technique has been proposed in [13] which models the attack graph first and then clusters the sequence of alerts. The technique is tested for clustering the attack graphs with a high degree of similarity and dissimilarity. In [14], a new technique to cluster similar alerts generated from IDS is described. The root cause forces the IDS to generate these alerts. Experiments show that the technique is able to reduce 74% of false alerts from the given number of alerts. A model has been proposed to cluster the alerts and calculate the alert distance to improve the alert quality, and the alerts are reduced up to 80.55% [15].

A framework which uses clustering and prioritizing alerts to reduce false positives and adds contextual information for true positives has been presented in [16]. In prioritization component, priority level is calculated for each meta-alert based on the similarities between each meta-alert. Clustering is done by DBSCAN algorithm,

and the attack pattern discovery component extracts a set of features from each cluster. The reporting system is an interface for the analyst to sort and filter alerts based on priority, size, etc., for experimentation, two scenarios are considered and in both scenarios, highly prioritized alerts have the least frequencies. This framework significantly reduces the number of alerts that a network analyst has to analyse through correlation and then by filtering out low priority meta-alerts. Using the reporting system, the network analyst can visualize the clusters and patterns of each meta-alert.

### 3 Proposed System

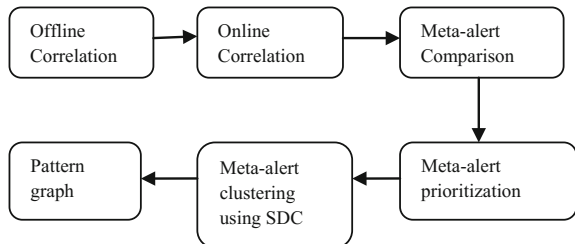
The proposed framework consists of six components: (1) offline correlation, (2) online correlation, (3) meta-alert comparison, (4) meta-alert prioritization, (5) meta-alert clustering and (6) pattern graph [16]. The training system considers attributes of alert such as timestamp, message type, source port, source IP address, destination port and destination IP address (Fig. 1).

In [16], offline correlation process fetches the historical alerts from the database. For every alert pair in the historical alert set, get constraints by calculating the probability and check it with the threshold value. Repeat the same for all combinations of attributes. Return the constraints and calculate minimum likelihood for all pairs of alerts. The constraint and correlation likelihood tables are generated in this process.

#### 3.1 Offline Correlation Algorithm

1. function OFFLINE
2. Fetch A, T (timestamp, message type, source port, source ip address, destination port, destination ip address) from the database
3. for all pairs of alerts in T
4. call function GETCONSTRAINTS

**Fig. 1** Proposed architecture





5. calculate minimum likelihood value for every pair of alerts for the given combination of constraints
6. end for
7. end function
  
1. function GETCONSTRAINTS
2. set constraint matrix as null
3. for all combinations of attributes
4. find the probability of alert1 and alert2 for the given attribute combination
5. if the probability is greater than the assigned threshold
6. then
7. Add that attribute combination to constraint
8. end if
9. end for
10. return constraint
11. end function

In online correlation process, incoming alerts are compared with the set of alerts captured in real time. Also the relevant correlation likelihood and constraint are calculated from the offline correlation process. If the correlation likelihood for incoming alerts and the alert sets is greater than or equal to threshold value and if at least one respective constraint is present, then the incoming alert can be correlated with the set of alerts. Only the correlated alerts are considered for further analysis, and the uncorrelated alerts are removed from the system.

### ***3.2 Online Correlation Algorithm***

1. function ONLINE
2. Fetch set of alerts occurred before the incoming alert
3. Compare the incoming alert with the set of alerts
4. Find the relevant constraint in the constraint Table
5. if correlation likelihood for incoming alert and alert set is greater than or equal to threshold
6. if respective constraint present in incoming alert and alert set
7. then
8. the incoming alert can be correlated
9. end if
10. end if
11. end function

The online correlation function reduces the volume of alerts and generates meta-alerts. The distance between each pair of meta-alerts will be calculated. This

distance measure will be used by meta-alert prioritization and meta-alert clustering functions. The distance between meta-alerts is based on the similarity between the attributes of those meta-alerts.

To evaluate the processing of alerts based on its severity, alert prioritization is used. The meta-alert prioritization function will assign a priority level to each meta-alert based on its similarity to a set of other meta-alerts. There are four priority levels, and meta-alerts which are highly similar to others are typically associated with a priority 1 or 2 while highly dissimilar meta-alerts are assigned level 3 or 4 [17]. Meta-alerts and distance values from meta-alert comparison will be used to assign priority to each of the meta-alerts. The degree to which a meta-alert is an outlier is calculated using the local outlier factor [17]. nLOF is calculated using meta-alert’s neighbourhood, reachability distance and local density.

Step 1: K-distance calculation: The K-distance of a meta-alert  $g_i$  is the distance between  $g_i$  and k-th nearest meta-alert. K is a configurable parameter for the algorithm’s computation. The following figure shows the clusters divided using K-means algorithm and the calculated K-distance.

K-neighbourhood calculation: A K-neighbourhood of a meta-alert  $g_i$ , denoted as  $N_k(g_i)$ , is a set of other meta-alerts in which the difference between any of the other meta-alerts and  $g_i$  is less than or equal to the K-distance. The following table shows the calculated K-neighbourhood distance for each meta-alert.

Step 2: Reachability distance: This is the maximum distance between two meta-alerts and latter meta-alert’s K-distance.

$$rdk(g, g_j) = \max\{ D(g, g_j), K - \text{distance}(g_j) \}$$

Step 3: Reachability distance calculation: A meta-alert’s local reachability density is the inverse of the average reachability distance between it and its K-neighbourhood.

$$lrdg_i = \frac{1}{\text{average}(\text{reachabilitydistance}, k - \text{neighbourhooddistance})}$$

Step 4: For each meta-alert, local outlier factor is calculated.

$$\frac{\sum_{g_j \in N_k(g_i)} \frac{lrdg_i}{lrdg_j}}{|N_{g_i}|}$$

Step 5: LOF priority: a priority value from 1 to 4 has been assigned to each meta-alert. If LOF is greater than 1.00, then the meta-alert is considered as false-positive and it need not be prioritized.

If LOF ranges from

- 0.00–0.25—priority 1
- 0.25–0.50—priority 2
- 0.50–0.75—priority 3
- 0.75–1.00—priority 4

The alerts generated by one or more IDS will be stored in a centralized database. These alerts are always mixed with irrelevant and duplicate alerts. This will increase the effort of the analyst while identifying successful alerts. So alert clustering is used to sort and group the alerts of same kind into a cluster. Density-based clustering techniques can detect the clusters of different shapes and sizes from large amount of data which contain noise and outliers [18]. Here, a set of points will be given in some space, it clusters the points which are density-reachable and marking the points that lie alone in low-density regions as outliers (whose nearest neighbours are far away).

DBSCAN clusters subset of the points of the database, which satisfies following two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-connected to any point of the cluster, then it is a part of the cluster as well.

There are two parameters used to quantify density of points for clusters and noise in a data set in DBSCAN. The first is epsilon (*eps*) which is a chosen distance for any two points to be considered being in a neighbourhood. The second is the minimum number of points (*MinPts*) needed in a neighbourhood. Given the values of *eps* and *MinPts*, DBSCAN algorithm makes use of density-reachability to extend a particular cluster. A cluster is formed by DBSCAN for all points that are density-reachable. The process of forming clusters and finding density-reachable points repeats until all points in the data set are examined.

In [5], SDC is proposed which is a part of density-based clustering that does not require a predefined number of clusters and also able to filter noise. SDC also uses *eps* and *MinPts*. SDC algorithm is applied in meta-alerts to group them into most suitable cluster. The algorithm is initialized by identifying small clusters with very high density as initial clusters. Rather than expanding the clusters by including other density-reachable points, SDC increases the radius of the identified clusters iteratively until it cannot further expand. SDC algorithm ensures that a required density must be reached in the initial clusters and uses scalable distances to expand the initial clusters.

### 3.3 Scalable Distance-Based Clustering Algorithm

1. Initialize all the meta-alerts as unclassified
2. Repeat
3. Randomly select a meta-alert as centroid
4. if number of points in eps-neighbourhood of  $p_i$  less than or equal to minimum points
5. then
6. create an initial cluster  $C_j$  by including centroid and all its eps- neighbourhood points
7. remove the centroid from the list of meta-alerts.
8. else
9.  $p_i$  is classified as X
10. remove  $p_i$  from the list of meta-alerts
11. until
12. list of meta-alerts become null
13. end if
14. For all initial cluster
15. repeat
16. Find the centroid
17.  $\text{eps} = \text{eps} - \Delta\text{eps}$
18. Add points from X in which the distance from the centroid of the cluster is larger than eps
19. Until no other points are found
20. The points remaining in X are considered noise
21. end for

The attack pattern discovery component receives the clusters of meta-alerts and attempts to extract a set of representative features for each cluster. This represents each meta-alert as a less complex graph structure. This graph structure is referred to as a pattern graph. A pattern graph is a graph representation of a meta-alert where each node represents meta-alert and the edges represent the relation between them.

## 4 Evaluation Metrics

### 4.1 Correlation

To measure the accuracy of the system, false-positive correlated rate and true-positive correlated rate have been considered [19]. The false-positive correlated rate represents the percentage of incorrectly correlated alert pairs among all the correlated pairs.

$$\text{False Positive Correlated Rate (FPC)} = \frac{\text{Number of false correlated pairs}}{\text{Number of correlated pairs}}$$

The true-positive correlated rate represents the percentage of true correlated pairs among the total number of alert pairs.

$$\text{True Positive Correlated Rate (TPC)} = \frac{\text{Number of true correlated pairs}}{\text{Number of related pairs}}$$

## 4.2 Prioritization

The TPR evaluates the ability of the system to correctly prioritize the meta-alerts. The false-positive rate evaluates the number of incorrectly prioritized meta-alerts among the total number of prioritized meta-alerts.

$$\text{True Positive Rate (TPR)} = \frac{\text{Number of correctly prioritized alerts}}{\text{Number of true positive alerts}}$$

$$\text{False Positive Rate (FPR)} = \frac{\text{Number of incorrectly prioritized alerts}}{\text{Number of prioritized alerts}}$$

## 4.3 Cluster Quality

### Purity:

Purity is used to measure accuracy by counting the number of correctly assigned alerts in the cluster and dividing by N.

$$\text{Purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega \cap C_j|$$

where N is the total number of elements distributed among the clusters,  $\Omega = \{ \omega_1, \omega_2, \dots, \omega_k \}$  is the set of clusters, and  $C = \{ C_1, C_2, \dots, C_j \}$  is the set of classes. The greater the value of purity indicates good clustering.

### Davies–Bouldin index:

Davies–Bouldin index is used to validate how well the cluster has been formed. Algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index. The clustering algorithm that produces a collection of

clusters with the smallest Davies–Bouldin index is considered the best algorithm based on this criterion.

$$\text{Davies – Bouldin index} = \frac{1}{n} \sum_{i=1}^n \frac{(\sigma_i + \sigma_j)}{d(C_i, C_j)}$$

where  $n$  is the number of clusters,  $C_x$  is the centroid of cluster  $x$ ,  $\sigma_x$  is the average distance of all elements in cluster  $x$  to centroid  $C_x$ , and  $d(C_i, C_j)$  is the distance between centroid  $C_i$  and  $C_j$ .

## 5 Experimental Results

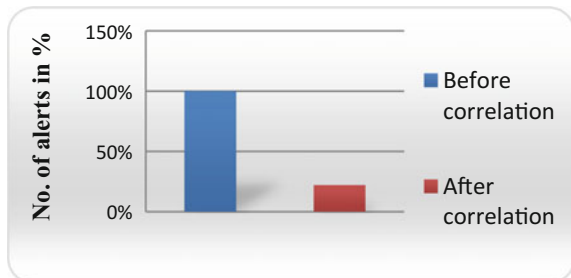
The historical alerts have been generated from DARPA 1999s week of data set. The true alerts and false alerts are analysed in the system. Correlation and prioritization functions reduce the number of alerts being analysed. The true-positive correlation rate is 68% and false-positive correlation rate is 33%. The true-positive rate and false-positive rate are evaluated as 0.297 and 0.48, respectively. The purity value calculated for DBSCAN algorithm is 0.75 and for SDC it is 0.91. This implies that SDC algorithm is performing more perfect clustering than DBSCAN algorithm, i.e. the SDC algorithm groups similar alerts into same cluster.

Figure 2 shows that after the correlation process, the false-positive alerts are reduced effectively by 22%. Then, the result of correlation process is given as input to prioritization.

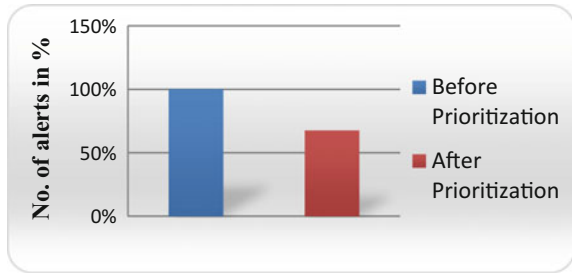
Figure 3 shows that the 22% of correlated alerts are further prioritized. Prioritization process assigns priority to all the alerts, and then the low priority alerts are eliminated. Thus, the false-positive alerts are further reduced to 67.50%. The prioritized alerts are then clustered using DBSCAN and SDC algorithms.

Figure 4 shows the comparison of DBSCAN algorithm and SDC algorithm. Total number of meta-alerts that are clustered are more in DBSCAN algorithm than

**Fig. 2** Reduction of false-positive alerts using correlation



**Fig. 3** Reduction of false-positive alerts in prioritization



**Fig. 4** Clustering results comparison



SDC algorithm. But in SDC algorithm, more number of meta-alerts are correctly clustered than using DBSCAN algorithm. SDC algorithm also detects large number of meta-alerts than DBSCAN algorithm. SDC algorithm eliminates more noise compared to DBSCAN algorithm.

## 6 Conclusion

This paper describes the need for correlation and prioritization which is used to minimize the number of alerts to be analysed by the network analyst. The experimental results show that correlation and prioritization reduces the number of false alerts. Clustering the prioritized meta-alerts further effectively mitigates the false alerts. Also, the performance between DBSCAN and SDC algorithms is compared and found that SDC algorithm effectively mitigates false alerts than DBSCAN algorithm.

## References

1. Debar, H., Dacier, M., Wespi, A.: Towards a taxonomy of intrusion detection systems. *Comput. Netw.* **31**(8), 805–822 (1999)
2. Sandhu, U.A., Haider, S., Naseer, S., Ateeb, O.U.: A survey of intrusion detection & prevention techniques. In: 2011 International Conference on Information Communication and Management IPCSIT, vol. 16 (2011)
3. Alsubhi, K., Al-Shaer, E., Boutaba, R.: Alert Prioritization in Intrusion Detection Systems
4. Lagzian, S.: Frequent item set mining-based alert correlation for extracting multi-stage attack scenarios. In: IEEE Telecommunications (IST), 2012 Sixth International Symposium, pp. 1010–1014 (2012)
5. Yang, C.C., Ng, T.D.: Analyzing and visualizing web opinion development and social interactions with density-based clustering. In: Proceedings of the International WWW Conference, pp. 1144–1155 (2011)
6. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.A.: A comprehensive approach to intrusion detection alert correlation. *IEEE Trans. Depend. Secur. Comput.* **1**, 146–169 (2004)
7. Yu, J., Ramana Reddy, Y.V., Selliah, S., Reddy, S., Vijayan, Bharadwaj, Kankanahalli, S.: TRINETR: an architecture for collaborative intrusion detection and knowledge based alert evaluation. *Adv.Eng. Inform. (Elsevier)* **19**, 93–101 (2005)
8. Lee, S., Chung, B., Kim, H., Lee, Y., Park, C., Yoon, H.: Real-time analysis of intrusion detection alerts via correlation. *Comput. Secur. (Elsevier)* **25**, 169–183 (2006)
9. Noel, S., Jajodia, S.: Optimal IDS sensor placement and alert prioritizing using attack graphs. *J. Netw. Syst. Manag.* **16**, 259–275 (2008)
10. Alsubhi, K., Al-Shaer, E., Boutaba, R.: Alert prioritization in intrusion detection systems. In: NOMS 2008–2008 IEEE Network Operations and Management Symposium, pp. 33–40 (2008)
11. Pietraszek, T., Tanner, A.: Data mining and machine learning—towards reducing false positives in intrusion detection. *Inf. Secur. Tech. Rep. (Elsevier)* **10**, 169–183 (2005)
12. Nikulin, V.: Threshold-based clustering with merging and regularization in application to network intrusion detection. *Comput. Statist. Data Anal. (Elsevier)* **51**, 1184–1196 (2006)
13. Patel, H.: *Intrusion Alerts Analysis Using Attack Graphs and Clustering*. Masters', San Jose State University (2009)
14. Al-Mamory, S.O., Zhang, H.: Intrusion detection alarms reduction using root cause analysis and clustering. *Comput. Commun. (Elsevier)* **32**, 419–430 (2009)
15. Njogu, H.W., Wei, L.J.: Using alert cluster to reduce IDS alerts. In: Proceedings of the Third IEEE International Conference on Computer Science and Information Technology, pp. 467–471 (2011)
16. Shittu, R., Healing, A., Ghanea-Hercock, R., Bloomfield, R., Rajarajan, M.: Intrusion alert prioritisation and attack detection using post-correlation analysis. *Comput. Secur. (Elsevier)* **50**, 1–15 (2015)
17. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM Sigmod International Conference on Management of Data, pp. 1–12 (2000)
18. Shah, G.H., Bhensdadia, C.K., Ganatra, A.P.: An empirical evaluation of density-based clustering techniques. *Int. J. Soft Comput. Eng. (IJSCE)*. **2**(1) (2012). ISSN: 2231-2307
19. Ren, H., Stakhanova, N., Ghorbani, A.A.: An online adaptive approach to alert correlation. In: Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), pp. 153–172 (2010)



# Performance Analysis of Clustering-Based Routing Protocols for Wireless Sensor Networks



B. Chandirika and N. K. Sakthivel

**Abstract** Wireless sensor network (WSN) is one of the new paradigms for sharing and disseminating data through a wide area. It is used for a huge range of real-time applications such as military surveillance, healthcare industry. The sensor nodes are very important to monitor the changes in the real-time environment. During the dissemination of data, there is a loss in the energy level of the sensor nodes. Hence, an energy-efficient routing technique is required to maximize the network lifetime. This research work analyzes three routing protocols, namely Fan-Shaped Clustering (FSC), Fuzzy-based Clustering, and Ring Routing (RR) protocols. From the experimental analysis, it is noted that the ring routing is highly efficient than the FSC and Fuzzy-based Clustering in terms of residual energy, number of alive nodes and dead nodes. The FSC yields better throughput and minimum end-to-end delay than the Fuzzy-based Clustering and ring routing.

**Keywords** Clustering • Energy efficiency • Fuzzy logic • Residual energy  
Ring routing • Routing technique • Wireless sensor network

## 1 Introduction

Wireless sensor network (WSN) is densely deployed in the hazardous places where battery recharge or replacement is nearly impossible. Once the network is established, the nodes start sending the information and there is an exponential drain in the power drain. Sometimes, the same data or information can be received by the neighboring sensor nodes and sink. This data redundancy results in the degradation of the performance of WSN in terms of bandwidth utilization, battery power usage,

---

B. Chandirika (✉)  
Bharathiar University, Coimbatore, Tamilnadu, India  
e-mail: chandirikab@rediffmail.com

N. K. Sakthivel  
Nehru College of Engineering and Research Centre, Thrissur, Kerala, India  
e-mail: nksakthivel@gmail.com

and delay. Hence, energy conservation is an important challenge in the WSN. To address the data redundancy issue and enhance the performance of routing protocol, various routing protocols have been proposed by the researchers. Clustering is one of the best strategies to reduce the energy consumption and maximize the network lifetime. In the clustering process, the sensor nodes are grouped as cluster. The nodes identified for coordinating all these clusters are called as Cluster Head (CH). Each and every node under a cluster senses the information and forwards the sensed information to the CH that aggregates all the received information and forwards to the sink. Thus, the network load and energy conservation can be reduced greatly by this model and network lifetime can be improved. This research work focused on a few recently proposed clustering techniques, namely FSC [1, 2], Fuzzy-based Clustering [3], and Ring routing [4].

The rest of the paper is systematized as follows. Section 2 describes the existing works about the clustering and routing protocols in WSN. Section 3 explains the FSC, Fuzzy-based Clustering, and Ring routing methodology. Section 4 illustrates the experimental setup and comparative analysis of these routing protocols. The proposed work is concluded in Sect. 5.

## 2 Related Works

Tyagi and Kumar [5] presented a survey about the clustering and routing techniques in WSN. The merits and drawbacks of the prominent techniques are highlighted to facilitate selection of a particular technique based on the advantages. Singh and Sharma [6] surveyed cluster-based routing protocols along with the merits and drawbacks. Pantazis et al. [7] provided an analytical-based review about various energy-efficient routing protocols in WSN. Jan et al. [8] proposed a clustering-based hierarchical routing protocol to improve the lifetime and data quality in WSN. Sharma et al. [9] analyzed the cluster-based routing protocol in WSN through the simulation process. Razaque et al. [10] discussed Hybrid-Low Energy Adaptive Clustering Hierarchy (H-LEACH) for solving the problems of energy concerns during selection of channel head. The residual and maximum energy of the nodes for every round are considered while selecting the channel head. Dey et al. [11] implemented a Fruit Fly (FF)-based clustering protocol to reduce the power consumption of the nodes and improve the network lifetime. Prusty et al. [12] proposed a Hybrid Multi-Hop cluster routing protocol for uniform distribution of energy among the nodes. Zahedi et al. [13] suggested a Fuzzy routing protocol based on the swarm intelligence for controlling the distribution of CHs over the network. The proposed protocol generated balanced clusters and improved network lifetime. Sabor et al. [14] presented a detailed classification of hierarchical-based routing protocols and investigated the comparison between these protocols. Hande et al. [15] suggested a new CH selection method to ensure the extended lifetime of the network. The network failure due to the death of CH is reduced. Shelke et al. [16]

presented an enhanced version of Fuzzy-based LEACH protocol that considers residual battery power, centrality of CH, and node density during routing.

### 3 Clustering Techniques

In this section, the three popular clustering-based routing protocols [3, 4, 17] are discussed.

#### 3.1 FSC Protocol

The FSC technique involves the following assumptions [17].

- The sensor nodes are uniformly and independently distributed in a sensor field.
- All sensor nodes have the same fixed transmission power and transmission rate.
- Each sensor node is aware of its polar coordinates to the sink, which can be obtained through Global Positioning System (GPS) or some other sensor localization techniques.

This model is designed to address the clustering of large-scale sensor networks. The area is split and partitioned into fan-shaped clusters. The sink is fixed at the center of the cluster. The fan-shaped cluster is partitioned as ring layers that may be concentric one. The main feature of this model is that the clusters are equal-sized. Hence, it facilitates to achieve load balancing.

#### 3.2 Fuzzy Clustering

Fuzzy logic-based clustering protocols involve selecting or electing a right and an efficient CH that enhances the overall network lifetime. Nayak et al. [3] proposed an efficient Fuzzy-based Clustering technique with Super Cluster Head (SCH) that overcomes the overhead of collecting and calculating the energy level of each sensor node. In this model, the three Fuzzy descriptions [3, 18], namely remaining battery power, mobility, and centrality, are considered to identify the best SCH and the newly nominated SCH will take responsibility to deliver a message to the Base Station (BS).

*for (EachRound)*

{

*Identify and Select CHs based on Threshold Value*

*Select  $K_{optimal}$  CHs in each round and choose best Node as SCH*

*$K_{optimal}()$*

```

{
All CHs send the aggregated data to SCH
}
BS collects the information from SCH
}

```

### 3.3 Ring Routing

Ring routing methodologies, namely route construction, advertisement, obtaining sink position from the ring, data dissemination, and ring range, are used for framing an efficient ring routing [4]. This protocol imposes three roles on the sensor nodes while constructing the ring after the deployment of the sensor network. They are ring node, regular node, and anchor node. The ring node will form a closed ring structure, and it will advertise the location of the ring. The regular nodes will collect the information about the position of the sink from the constructed ring. The anchor node is introduced to make a connection between the sink and sensor networks to disseminate the collected information. The radius of an initial ring is determined. A ring node is initially selected at random by the defined radius, and a complete closed loop corresponding to the network center is to be formed, so that it can reach all the nodes within the ring. Otherwise, there is a need to select another node and repeat the procedure until the starting node is reached and the closed loop is formed. After a certain number of repetitions, if the ring cannot be formed, the size of the radius can be changed and the procedure can be repeated.

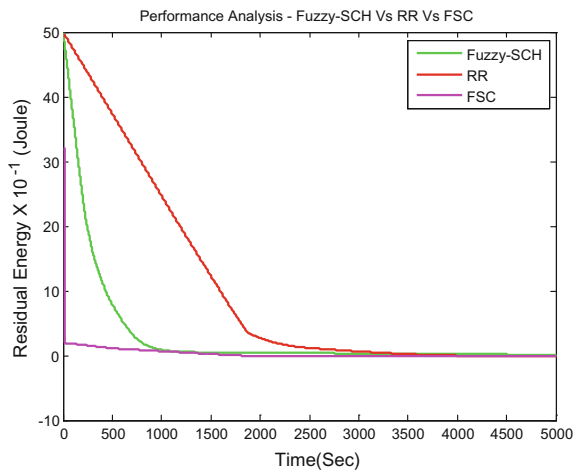
## 4 Performance Analysis

The three routing protocols are implemented in VC++ and integrated with QualNet 5.0 R Simulator and analyzed. The network field is defined as a circular area where the nodes are randomly deployed. The sink is assumed to be located at the center of the area, and it is allowed to move randomly. Let us assume each node can send one data packet periodically, i.e., 1 s (one round). The transmission radius of the nodes is  $100 \times 100$  m, and width of the layers  $r$  and  $r'$  will be 40 m each. The initial energy of nodes will be 2 J, and minimum energy threshold will be 0.2 J. It is designed to create around 5 clusters, and bandwidth for applications is limited to 1 MBPS. For each cluster, the setup is executed more than 5000 s.

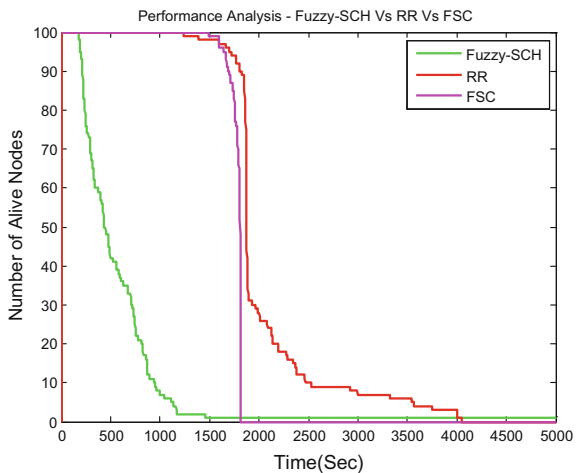
The recently proposed three routing protocols are analyzed thoroughly in terms of residual energy, number of alive nodes, number of dead nodes, throughput, and end-to-end delay. From Fig. 1, it is noted that the residual energy of the ring routing protocol is relatively high as compared with the FSC and Fuzzy-based Clustering techniques. The lifetime of sensor network can be maximized by the Ring routing

protocol. In the ring routing protocol, more alive nodes are available even after 5000 s for data dissemination. This leads to the decrease in the dead nodes as shown in Figs. 2 and 3. Hence, the ring routing is an energy-efficient routing technique as compared with the FSC and Fuzzy-based Clustering. As the large-scale network is divided and split as number of FSC for communication, the complexity for data dissipation is relatively less in the FSC as compared with Fuzzy-based Clustering and Ring routing protocols. This facilitates to forward more packets during communication. Figures 4 and 5 depict the performance analysis in terms of throughput and end-to-end delay. As a result, the FSC technique achieves higher throughput and minimum end-to-end delay as compared with the Fuzzy-based Clustering and Ring routing protocols.

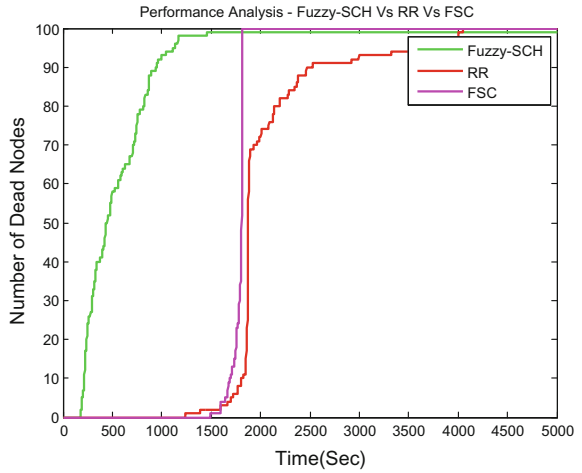
**Fig. 1** Performance analysis in terms of residual energy



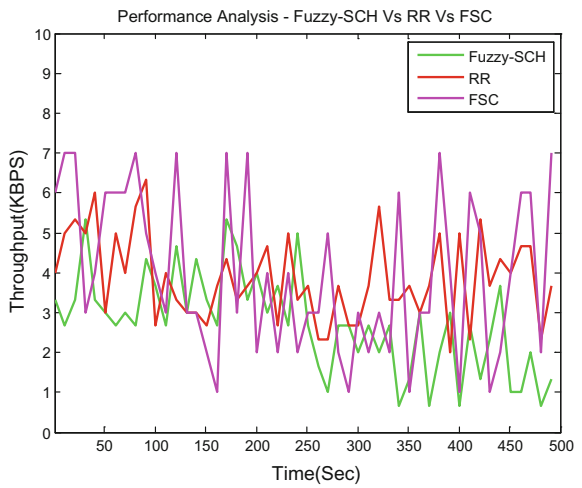
**Fig. 2** Performance analysis in terms of alive nodes



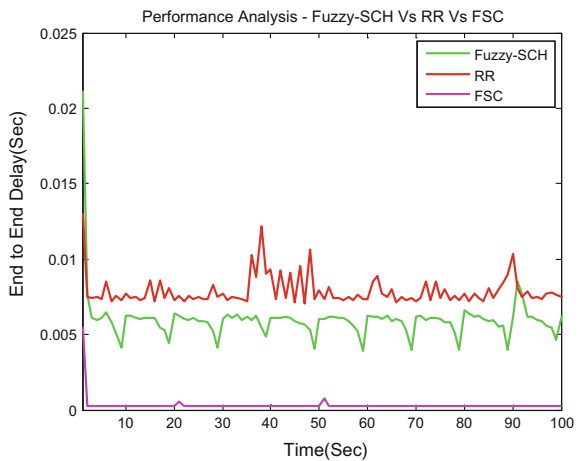
**Fig. 3** Performance analysis in terms of dead nodes



**Fig. 4** Performance analysis in terms of throughput



**Fig. 5** Performance analysis in terms of end-to-end delay



## 5 Conclusion

This research work involves the comparative analysis of three routing protocols. These routing techniques have been implemented with QualNet 5.0R Simulator and analyzed thoroughly in terms of residual energy, number of alive nodes, the number of dead nodes, throughput, and end-to-end delay. From the analysis, it is noted that the ring routing is an energy-efficient routing technique as compared with the FSC and Fuzzy-based Clustering in terms of residual energy, number of alive nodes, and the number of dead nodes. It is also revealed that the FSC performs well as compared with the Fuzzy-based Clustering and Ring routing protocols in terms of throughput and end-to-end delay.

## References

1. Di Francesco, M., Das, S.K., Anastasi, G.: Data collection in wireless sensor networks with mobile elements: a survey. *ACM Trans Sensor Netw (TOSN)* **8**, 7 (2011)
2. Yin, F., Li, Z., Wang, H.: Energy-efficient data collection in multiple mobile gateways WSN-MCN convergence system. In: 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC), pp. 271–276 (2013)
3. Nayak, P., Devulapalli, A.: A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime. *IEEE Sens. J.* **16**, 137–144 (2016)
4. Tunca, C., Isik, S., Donmez, M.Y., Ersoy, C.: Ring routing: an energy-efficient routing protocol for wireless sensor networks with a mobile sink. *IEEE Trans. Mob. Comput.* **14**, 1947–1960 (2015)
5. Tyagi, S., Kumar, N.: A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *J. Netw. Comput. Appl.* **36**, 623–645 (2013)
6. Singh, S.P., Sharma, S.: A survey on cluster based routing protocols in wireless sensor networks. *Procedia Comput. Sci.* **45**, 687–695 (2015)
7. Pantazis, N.A., Nikolidakis, S.A., Vergados, D.D.: Energy-efficient routing protocols in wireless sensor networks: a survey. *IEEE Commun. Surveys Tut.* **15**, 551–591 (2013)
8. Jan, M.A., Nanda, P., He, X., Liu, R.P.: Enhancing lifetime and quality of data in cluster-based hierarchical routing protocol for wireless sensor network. In: IEEE 10th International Conference on High Performance Computing and Communications and 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC\_EUC), pp. 1400–1407 (2013)
9. Sharma, A., Agrawal, M., Sondhi, M., Singh, A.K.: Analysis and simulation of energy efficient optimal scenarios for cluster based routing protocol in wireless sensor network. *J. Netw. Security Comput. Netw.* **2** (2016)
10. Razaque, A., Mudigulam, S., Gavini, K., Amsaad, F., Abdulgader, M., Krishna, G.S.: H-LEACH: hybrid-low energy adaptive clustering hierarchy for wireless sensor networks. In: IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1–4 (2016)
11. Dey, A., Sarkar, T., Ali, S.: Fruit Fly algorithm based clustering protocol in wireless sensor networks. In: 9th International Conference on Electrical and Computer Engineering (ICECE), pp. 295–298 (2016)
12. Prusty, A.R., Sethi, S., Nayak, A.K.: A hybrid multi-hop mobility assisted heterogeneous energy efficient cluster routing protocol for wireless ad hoc sensor networks. *J. High Speed Netw.* **22**, 265–280 (2016)

13. Zahedi, Z.M., Akbari, R., Shokouhifar, M., Safaei, F., Jalali, A.: Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks. *Expert Syst. Appl.* **55**, 313–328 (2016)
14. Sabor, N., Sasaki, S., Abo-Zahhad, M., Ahmed, S.M.: A comprehensive survey on hierarchical-based routing protocols for mobile wireless sensor networks: review, taxonomy, and future directions. *Wireless Commun. Mobile Comput.* **2017** (2017)
15. Hande, Y., Tripathy, S., Phalke, K.: Energy efficient clustering based routing protocol in WSN. *Int. J. Eng. Sci.* **2109** (2016)
16. Shelke, M., Tefera, G., Malhotra, A., Mahalle, P.: Fuzzy-based fault-tolerant low-energy adaptive clustering hierarchy routing protocol for wireless sensor network. *Int. J. Wireless Mobile Comput.* **11**, 117–123 (2016)
17. Lin, H., Wang, L., Kong, R.: Energy efficient clustering protocol for large-scale sensor networks. *IEEE Sens. J.* **15**, 7150–7160 (2015)
18. Nehra, V., Pal, R., Sharma, A.K.: Fuzzy-based leader selection for topology controlled pegasis protocol for lifetime enhancement in wireless sensor network. *Int. J. Comput. Technol.* **4**, 755–764 (2013)



# A Secure Encryption Scheme Based on Certificateless Proxy Signature



K. Sudharani and P. N. K. Sakthivel

**Abstract** Certificateless public key cryptography (CL-PKC) scheme is introduced for solving the key escrow problems in the identity-based cryptography and eliminating the use of security certificates. By introducing the proxy signature concept in the certificateless cryptography scheme, this certificateless proxy signature (CLPS) scheme has attracted the attention of more researchers. However, this scheme suffers due to the security issues and fails to achieve the unforgeability against the attacks. To overcome the security issues in the existing cryptographic schemes, this paper proposes an encryption scheme based on the certificateless proxy signature for sharing the sensitive data in the public cloud in a secure manner. The proposed scheme is proven to be unforgeable against the message attacks. When compared with the existing CLPS scheme without random oracles, the proposed scheme offers better data security while ensuring better data sharing performance. From the experimental results, it is noticed that the proposed scheme requires minimum encryption time and decryption time than the existing schemes.

**Keywords** Access control • Cloud computing • Certificateless public key cryptography (CL-PKC) • Data confidentiality • Malicious KGC attack Proxy signature • Public key replacement attack

## 1 Introduction

The CL-PKC scheme was introduced by integrating the partial private key and a secret value chosen by the user. This scheme addresses the key escrow issues in the IB-PKC scheme. The certificateless proxy re-encryption (CL-PRE) scheme [1]

---

K. Sudharani (✉)  
Bharathiar University, Coimbatore, Tamilnadu, India  
e-mail: ksudharani.shagthi@gmail.com

P. N. K. Sakthivel  
Nehru College of Engineering and Research Centre, Thrissur, Kerala, India  
e-mail: nksakthivel@gmail.com

depends on the pairing operations for solving the key escrow problems and certificate management issues. The pairing operations require high computational cost than the modular exponentiation. Z. Eslamiet al. [2] developed the proxy signature concept that permits an entity called as an original signer to assign the signing authority to the proxy signer. But, the security of this scheme is not proven formally. The mediated certificateless public key encryption (mCL-PKE) scheme involves the data owner, public cloud, and users. The owner requests the cloud to partially decrypt the encrypted data, when the users requesting the confidential data from the owner. The mCL-PKE approach ensures efficient key generation and management functionality deployed in the untrusted cloud, without any key escrow problem. The key generation center (KGC) cannot acquire the private keys of the users. To enhance the efficiency of the mCL-PKE scheme, the security intermediary (SI) stores the partially decrypted data in the cloud storage, after completing the initial partial decryption for each user. If a user is revoked, the owner updates the access control list at the SI, to deny the future access requests from the revoked user. During the addition of a new user, the owner performs data encryption using the public key of the user and uploads the data and updated access control list to the cloud. The existing users are not affected by the user revocation and the addition of new users.

The remaining sections of the paper are systematized as follows: Section 2 describes the improved secure cloud storage. Section 3 explains the proposed secure encryption scheme. Section 4 provides the comparative analysis of the proposed and existing cryptographic schemes. Section 5 concludes the proposed work.

## 2 Related Works

Wang et al. [3] proposes a certificateless proxy re-encryption (CL-PRE) scheme without pairings. This scheme is efficient and requires minimum computational cost. This scheme solves the key escrow problem and does not require a public key certificate. Qin et al. [4] presents a secure CL-PRE scheme without using the bilinear pairing. The proposed scheme is highly secure against the adaptive chosen ciphertext attack. Srinivasan and Rangan [5] constructed the unidirectional CCA-secure CL-PKE scheme without pairing by extending the public key infrastructure (PKI). Seo et al. [6] developed an efficient certificateless encryption scheme without the pairing operations. The encryption efficiency at the data owner side is improved. Xu et al. [7] created a generally hybrid PRE (GHPRE) scheme to achieve secure sharing of data between PRE and public key encryption (PKE) schemes. Lu and Li [8] proposed a certificate-based PRE scheme for secure sharing of sensitive data in public clouds. The computational cost and time consumption are reduced by preventing the need for the bilinear pairing operations. Sur et al. [9] suggested a certificate-based PRE scheme for improving data confidentiality in the public cloud storage. Lu [10] proposed a novel certificate-based PRE scheme for sharing the

encrypted data in public clouds. A parallel CL-PRE (PCL-PRE) is proposed by applying parallel computing into the PRE algorithm to ensure safe access control of data in cloud storage [11]. Kumar et al. [12] presented a mediated CL-PKE (mCL-PKE) scheme without pairing operations.

### 3 Proposed Scheme

Most of the CL-PKC schemes require high computational cost, as they are based on the bilinear pairings. The computational complexity is minimized by using a pairing-free approach. The computational cost for decryption is reduced, since a semi-trusted security intermediary (SI) partially decrypts the encrypted data before the data decryption by the users. The proposed encryption scheme is efficient than the pairing-based scheme. It can perform efficient key management and user revocation, when compared to the symmetric key-based mechanisms. In the symmetric key-based cryptographic mechanisms, the users should manage the number of keys equal to the logarithmic value of the number of users, and the private keys should be updated during the revocation of the users. In our encryption scheme, there is no need to change the private keys of the users. Our scheme also guarantees the privacy of data stored in the public clouds while implementing the access control requirements. The SI, KGC, and storage service are semi-trusted and exist in a public cloud. They are trusted completely for the data and key confidentiality; they are trusted for the accurate execution of the protocols. The owner uploads the encrypted data to the cloud.

#### Advantages of proposed scheme

The main advantage of this scheme is the KGC resides in a cloud, when compared to the conventional approaches. It simplifies the key management task for the business enterprises. The KGC generates a partial private key. The user chooses the private key including a secret value. In our scheme, the partial private key is provided to the SI in a secure way and the user maintains the secret value as an own private key. Therefore, the access request received from each user is passed through the SI to verify whether the user is revoked before the partial decryption of the encrypted data using the partial private key. Hence, the proposed encryption scheme does not suffer from the key escrow problem, as the private key of the user is not revealed to any third party. Neither the KGC nor the SI can decrypt the data for the specific users. As the access request is processed through the SI, the proposed scheme supports the immediate revocation of compromised users. Our scheme is non-forgeable against the chosen-message attacks. The advantages are as follows:

- It solves the security flaws in the existing CLPS scheme [13] by resisting the public key replacement and malicious KGC attacks.
- It offers high security guarantee and requires minimum computational cost.

**Setup [14]:** Let  $G_1$  and  $G_2$  are the two cyclic multiplicative groups of prime order 'p', 'g' is a generator of  $G_1$  and 'e'.  $G_1 \times G_1 \rightarrow G_2$  is a bilinear map. The KGC selects a random value  $\alpha \in Z_p^*$  and computes  $g_1 = g^\alpha$ . It also chooses  $g_2, u', u_1, \dots, u_n, v_0, v_1, m_0, m_1 \in G_1$  in a random way. Furthermore, the collision-resistant hash functions are  $H_0: \{0, 1\}^* \rightarrow \{0, 1\}^n$ ,  $H_1: \{0, 1\}^* \rightarrow Z_p^*$ , and  $H_2: \{0, 1\}^* \rightarrow Z_p^*$ . Let 'Q' be a point in the group  $G_1$  which is defined as follows. If the X-coordinate of the point 'Q' is odd, then  $f(Q) = 1$ . Otherwise,  $f(Q) = 0$ . The public parameters are  $P = \{G_1, G_2, e, p, g, g_1, g_2, u', u_1, \dots, u_n, v_0, v_1, m_0, m_1, H_0, H_1, H_2, f\}$ , and the master key is  $msk = g_1^\alpha$ .

**Partial Private Key Generation:** To generate a partial private key for a user 'U' with identity  $ID_U$ , the KGC randomly selects  $r_U \in Z_p^*$  and computes

$$psk_U = (psk_{U1}, psk_{U2}) = \left( g_1^\alpha \cdot \left( u' \prod_{i=1}^n u_i^{\theta_{U,i}} \right)^{r_U} \right) \cdot g^{r_U} \quad (1)$$

where  $\theta_{U,i}$  denotes the  $i$ th bit of  $\theta_U = H_0(ID_U)$ .

**Set Secret Value:** The user 'U' randomly chooses  $x_U \in Z_p^*$  and sets the secret value as  $SV_U = x_U$ .

**Set Public Key:** User 'U' computes the public key

$$PK_U = (PK_{U1}, PK_{U2}, PK_{U3}) = \left( g_1^{x_U}, g_2^{1/x_U}, e(g_1, g_1)^{x_U^2} \right) \quad (2)$$

The validity of the public key is verified by checking the sustainability of equations  $e(PK_{U1}, PK_{U2}) = e(g_1, g_2)$  and  $e(PK_{U1}, PK_{U1}) = PK_{U3}$ .

**Set Private Key:** The user 'U' chooses  $r'_U \in Z_p^*$  in a random way and computes

$$SK_U = (SK_{U1}, SK_{U2}) = \left( psk_{U1}^{x_U^2} \cdot \left( u' \prod_{i=1}^n u_i^{\theta_{U,i}} \right)^{r'_U} \cdot psk_{U2}^{x_U^2} \cdot g^{r'_U} \right) \quad (3)$$

Let  $\tilde{r}_U = r_U x_U^2 + r'_U$ , we have

$$SK_U = \left( g_1^{\alpha^2 x_U} \cdot \left( u' \prod_{i=1}^n u_i^{\theta_{U,i}} \right)^{\tilde{r}_U}, g^{\tilde{r}_U} \right) \quad (4)$$

**Delegation Generation:** To compute a delegation certificate for a warrant message  $W_{OP}$ , an original signer 'O' randomly chooses  $s \in Z_p^*$  and computes

$$DC_{OP} = (DC_{OP1}, DC_{OP2}, DC_{OP3}) = \left( g^s, SK_{O2}, SK_{O1} \cdot (PK_{O2}^\gamma \cdot v_\lambda)^s \right) \quad (5)$$

$$\text{Where } \lambda = f(DC_{OP2}) \text{ and } \gamma = H_1(DC_{OP1}, DC_{OP2}, ID_O, PK_O, W_{OP}, v_\lambda) \quad (6)$$

**Delegation Verification:** To verify a delegation certificate  $DC_{OP} = (DC_{OP1}, DC_{OP2}, DC_{OP3})$  for a warrant message  $w_{OP}$ , a proxy signer 'P' checks for the following equality

$$e(DC_{OP3}, g) = e(PK_{O1}, PK_{O1}) \cdot e\left(u' \prod_{i=1}^n u_i^{\theta_{O,i}}, DC_{OP2}\right) \cdot e(PK_{O2}^\gamma \cdot v_\lambda, DC_{OP1}) \quad (7)$$

where  $\theta_{O,i}$  denotes the  $i$ th bit of  $\theta_O = H_0(ID_O)$ ,  $\lambda = f(DC_{OP2})$ , and  $\gamma = H_1(DC_{OP1}, DC_{OP2}, ID_O, PK_O, w_{OP}, v_\lambda)$ . If it holds, the output is one. Otherwise, the output is zero.

**Proxy Signature Generation:** To sign a message 'M', the proxy signer 'P' randomly chooses  $s', t \in Z_p^*$ . The proxy signature  $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$  is computed as

$$\sigma = \left(g^t, SK_{P2}, DC_{OP1} \cdot g^{s'}, DC_{OP2}, DC_{OP3} \cdot (PK_{O2}^\gamma \cdot v_\lambda)^{s'} \cdot SK_{P1} \cdot (PK_{P2}^\eta \cdot m_\mu)^t\right) \quad (8)$$

where  $\lambda = f(DC_{OP2})$  and  $\gamma = H_1(DC_{OP1}, DC_{OP2}, ID_O, PK_O, w_{OP}, v_\lambda)$ ,  $\mu = f(\sigma_2)$  and  $= H_2(\sigma_1, \sigma_2, \sigma_3, \sigma_4, ID_O, PK_O, PK_P, M, m_\mu)$ .

**Proxy Signature Verification:** To verify the proxy signature, it is checked whether the following equality holds

$$e(\sigma_5, g) = PK_{O3} \cdot e\left(u' \prod_{i=1}^n u_i^{\theta_{O,i}}, \sigma_4\right) \cdot e(PK_{O2}^\gamma \cdot v_\lambda \cdot \sigma_3) \cdot PK_{P3} \cdot e\left(u' \prod_{i=1}^n u_i^{\theta_{P,i}}, \sigma_2\right) \cdot e(PK_{P2}^\eta \cdot m_\mu \cdot \sigma_1) \quad (9)$$

where  $\theta_{O,i}$  represents the  $i$ th bit of  $\theta_O = H_0(ID_O)$  and  $\theta_{P,i}$  indicates the  $i$ th bit of  $\theta_P = H_0(ID_P)$ ,  $\lambda = f(DC_{OP2})$  and  $\gamma = H_1(DC_{OP1}, DC_{OP2}, ID_O, PK_O, w_{OP}, v_\lambda)$ ,  $\mu = f(\sigma_2)$  and  $= H_2(\sigma_1, \sigma_2, \sigma_3, \sigma_4, ID_O, PK_O, PK_P, M, m_\mu)$ . If it holds, output is one. Otherwise, the output is zero.

## 4 Performance Analysis

The proposed secure encryption scheme based on certificateless proxy signature (SE-CLPS) is compared with the efficient certificateless short signature scheme (E-CLSS) [15], CLSS schemes developed by Choi et al. [16], Fan et al. [17], Tsoet al. [18], Chenet al. [19], and Heet al. [20]. Figure 1 shows the signature generation and signature verification time analysis of the proposed SE-CLPS and existing cryptographic schemes. The proposed scheme requires minimum time for signature generation and signature verification than the existing schemes. Figure 2

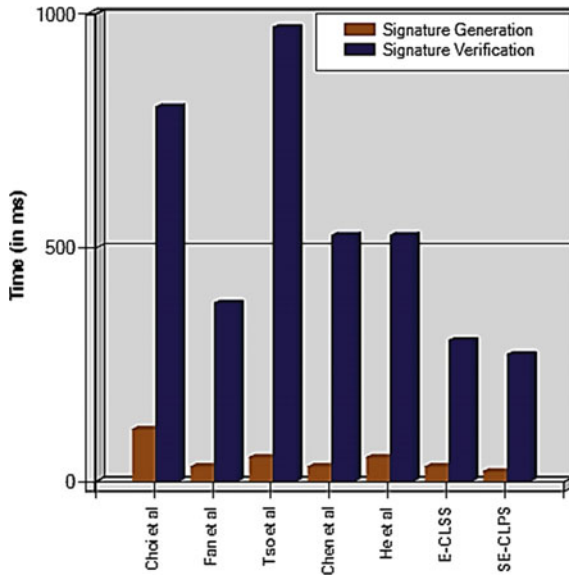


Fig. 1 Computational cost of proposed SE-CLPS and existing cryptographic schemes

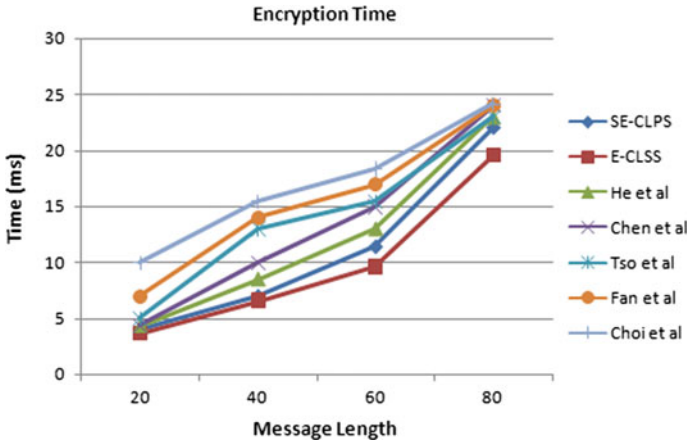
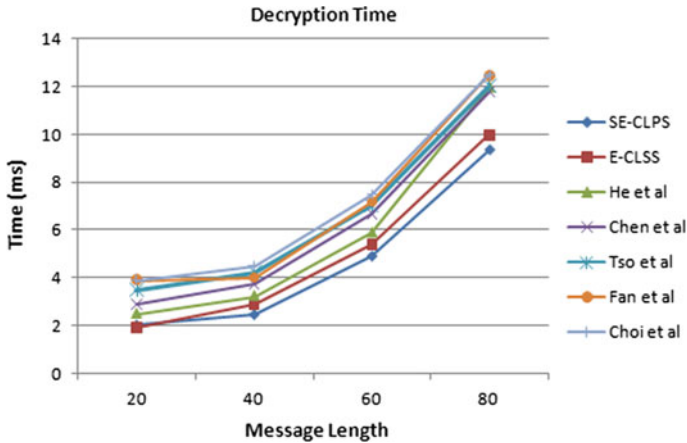


Fig. 2 Encryption time analysis

shows the time required for the encryption process in the proposed scheme for different message sizes. Since the proposed scheme does not use the pairing operations, it performs encryption efficiently. From the graph, it is observed that there is a linear increase in the encryption time with respect to the increase in the message size. There is a nonlinear increase in the cost with respect to the increase in the bit length, as the encryption algorithm performs the exponential operations.



**Fig. 3** Decryption time analysis

The similar observation is applied to the SI decryption and user decryption. The proposed scheme is implemented, where the data owner performs data encryption only once and creates a set of intermediate keys that enable the authorized users to decrypt the data. Our proposed scheme requires minimum encryption time than the existing schemes. In Fig. 3, we compare the time to perform decryption in the basic scheme and the improved scheme corresponding to the increase in the message size. The encryption time and decryption time increase with the increase in the number of users who can access the same data. From the graph, it is evident that the proposed encryption scheme yields better performance in comparison with the basic scheme, while allowing more users to access the same data item. The cost of the basic scheme is high as the encryption algorithm is executed for each user. From the graphs, we observe that the proposed scheme requires lower encryption time and decryption time than the existing cryptographic schemes.

## 5 Conclusion

This paper presented a secure certificateless proxy signature-based encryption scheme to securely share the sensitive data in public clouds. The proposed encryption scheme solves the key escrow and user revocation issues. Our proposed scheme supports immediate revocation of user and assures the confidentiality of the sensitive data stored in an untrusted public cloud while employing the access control policies of the data owner. The experimental results prove the efficiency of the proposed scheme than the existing cryptographic schemes. For multiple users satisfying the same access control policies, our proposed scheme encrypts data item

only once and reduces the overall computational overhead at the data owner. From the performance analysis, it is concluded that the proposed scheme achieves minimum encryption time and decryption time than the existing cryptographic schemes.

## References

1. Xu, L., Wu, X., Zhang, X.: CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, pp. 87–88 (2012)
2. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures: delegation of the power to sign messages. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **79**, 1338–1354 (1996)
3. Wang, L.-L., Chen, K.-F., Mao, X.-P., Wang, Y.-T.: Efficient and provably-secure certificateless proxy re-encryption scheme for secure cloud data sharing. *J. Shanghai Jiaotong Univ. (Sci.)* **19**, 398–405 (2014)
4. Qin Z, Wu S, Xiong H.: Strongly secure and cost-effective certificateless proxy re-encryption scheme for data sharing in cloud computing. In: International Conference on Big Data Computing and Communications, pp. 205–216 (2015)
5. Srinivasan, A., Rangan, C.P.: (2015) Certificateless proxy re-encryption without pairing: revisited. In: Proceedings of the 3rd International Workshop on Security in Cloud Computing, pp. 41–52
6. Seo, S.-H., Nabeel, M., Ding, X., Bertino, E.: An efficient certificateless encryption for secure data sharing in public clouds. *IEEE Trans. Knowl. Data Eng.* **26**, 2107–2119 (2014)
7. Xu, P., Xu, J., Wang, W., Jin, H., Susilo, W., Zou, D.: Generally hybrid proxy re-encryption: a secure data sharing among cryptographic clouds. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pp. 913–918 (2016)
8. Lu, Y., Li, J.: A pairing-free certificate-based proxy re-encryption scheme for secure data sharing in public clouds. *Future Gener Comput Syst* **62**, 140–147 (2016)
9. Sur, C., Park, Y., Shin, S.U., Rhee, K.H., Seo, C.: Certificate-based proxy re-encryption for public cloud storage. In: Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 159–166 (2013)
10. Lu, Y.: Efficient Certificate-Based Proxy Re-encryption Scheme for Data Sharing in Public Clouds (2015)
11. Jiang, W.: Parallel certificateless proxy reencryption scheme applicable for cloud storage. *J. Comput. Theor. Nanosci.* **13**, 4515–4520 (2016)
12. Kumar, A., Mishra, S., Dubey, P., Kumar, N.: Secure data sharing with data integrity in public clouds using mediated certificate-less encryption. In Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing, pp. 501–510 (2016)
13. Eslami, Z., Pakniat, N.: A certificateless proxy signature scheme secure in standard model. In Proceedings of International Conference on Latest Computational Technologies-ICLCT, pp. 81–84 (2012)
14. Lu, Y., Li, J.: Provably secure certificateless proxy signature scheme in the standard model. *Theore. Comput. Sci.* (2016)
15. Tsai, J.-L.: A new efficient certificateless short signature scheme using bilinear pairings. *IEEE Syst. J.* (2015)
16. Choi, K.Y., Park, J.H., Lee, D.H.: A new provably secure certificateless short signature scheme. *Comput. Math Appl.* **61**, 1760–1768 (2011)
17. Chun-Ifan, R.-H.H., Ho, P.-H.: Truly non-repudiation certificateless short signature scheme from bilinear pairings. *J. Inf. Sci. Eng.* **27**, 969–982 (2011)



18. Tso, R., Huang, X., Susilo, W.: Strongly secure certificateless short signatures. *J. Syst. Softw.* **85**, 1409–1417 (2012)
19. Chen, Y.-C., Horng, G., Liu, C.-L.: Strong non-repudiation based on certificateless short signatures. *IET Inf. Secur.* **7**, 253–263 (2013)
20. He, D., Huang, B., Chen, J.: New certificateless short signature scheme. *IET Inf. Secur.* **7**, 113–117 (2013)

# A Two-Stage Queue Model for Context-Aware Task Scheduling in Mobile Multimedia Cloud Environments



Durga S, Mohan S and J. Dinesh Peter

**Abstract** Multimedia cloud is an emerging computing paradigm that can effectively process media services and provide adequate quality of service (QoS) for multimedia applications from anywhere and on any device at lower cost. However, the mobile clients are still not getting their services in full due to its intrinsic nature such as limited battery life, disconnection, and mobility. In this paper, we propose a context-aware task scheduling algorithm that efficiently allocates the suitable resources to the clients. A queuing-based system model is presented with heuristic resource allocation. The simulation results showed that the proposed solutions provide better performance as compared to the state-of-the-art approaches.

**Keywords** Mobile cloud • Multimedia cloud • Queuing model  
Cuckoo search • Resource provisioning • Context awareness

## 1 Introduction

During the past decade, the benefits of hosting multimedia applications in the cloud are becoming increasingly attractive to both people and organizations. Recently, multimedia cloud computing [1] witnessed that there are few novel services, such as

---

D. S (✉) · J. D. Peter  
Karunya University, Coimbatore, India  
e-mail: durga.sivan@gmail.com

J. D. Peter  
e-mail: dineshpeter@karunya.edu

M. S  
CCIS, Al Yamamah University, KSA, Riyadh, Saudi Arabia  
e-mail: s.mohan77@gmail.com

cloud-based online photo and video editing, photo and video sharing, online gaming, video surveillance are springing up. A key problem in the multimedia cloud is to deal with the diverse and continuous changes of mobile context in fulfilling the QoS demands of users. In this paper, we tackle the aforementioned problems with the proposed adaptive resource provisioning algorithm which includes context awareness-based request scheduler and resource allocator. Queuing theory is used to model the proposed system model and cuckoo search [2] based optimization algorithm is presented that can allocate suitable physical resources for the request.

The rest of the paper is organized as follows: Related work is reviewed in Sect. 2. Section 3 describes the model description, mathematical formulation, and an optimized resource allocation algorithm. Section 4 presents the insight into implementation, performance evaluation, and discussion. Finally, Sect. 5 concludes the paper with future directions.

## 2 Related Work

Recent years have seen various proposals for frameworks and techniques for dynamic resource provisioning (DRP) in multimedia cloud environments. We identify two common DRP techniques, that we believe merit special attention, such that SLA-based RP (SBRP), deadline-based RP (DBRP). The main aspect of SBRP [3, 4] is to satisfy SLAs, the cloud provider had agreed with cloud users regarding the quantitative terms of functional and non-functional aspects of the service being offered, whereas in the DBRP [5, 6], the deadline for application completion, the time left for the deadline, and the average execution time of tasks that compose an application are considered to determine the no. of resources required by it. In 2013, Park et al. [7] presented a two-phase mobile device group creation mechanism for resource provisioning in mobile cloud for big data applications, where groups are created for  $n$  cut-off points of entropy values. In 2014, Sood et al. [8] discussed an adaptive model, in which an independent authority is employed to predict and store resources using ANN. However, the processing time and the communication cost involved with the independent authority are not considered. Biao et al. [9] presented a two-stage resource allocation strategy for handling multimedia tasks in the cloud. Nevertheless, this paper only focused on the minimization of the response time and cost. In our previous work [10], the cuckoo-based optimized resource allocation has been tested.

In the above-mentioned RP techniques, the priority of the workloads generated by the end users can be decided based on memory requirements, time requirements,

or any other resource requirements of the workload [11]; i.e., the workloads are handled in such a way that, the size of the workload, deadline of each task that fulfill the workload, available resources, and other management objectives are considered. But the local context information that affects the service provision cannot be influenced by the resource management decisions of the cloud provider.

### 3 System Model

#### 3.1 Proposed System Architecture

The proposed context-aware task scheduling and allocation model is shown in Fig. 1, which is originated from the traditional client-server model, where the mobile device acts as a service consumer. This model consists of two queues, which are request handler (RH) queue and media processing servers (MS) queue. The RH queue is maintained for scheduling the requests to the corresponding pool of media servers. Each pool of media servers is assigned with a MS queue which is responsible for optimal allocation of the resources. In multimedia cloud, the resource requests, to handle media tasks are translated into virtual machine resource requests, which in turn mapped into the allocation of suitable physical servers

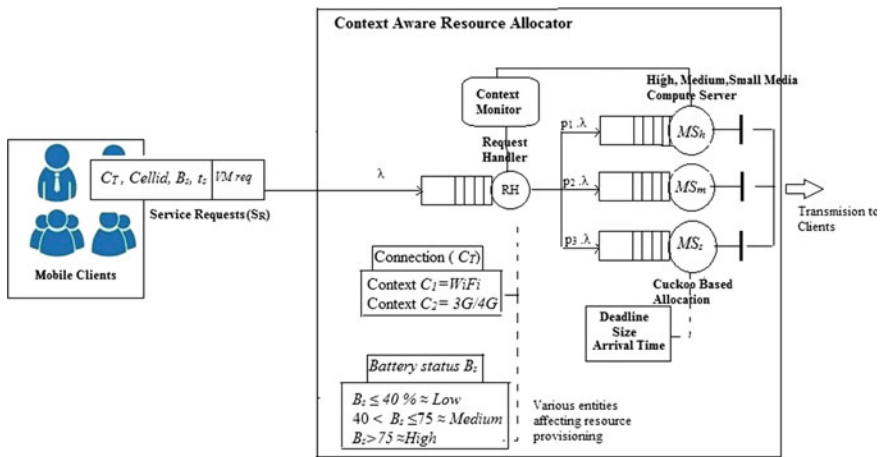


Fig. 1 Proposed resource allocation model

hosting the VMs. Majority of the clouds are constructed in the form of datacenter. In this model, we consider the data center architecture is built as large, medium, and small media server pools as per its compute, storage, memory, and bandwidth capacities and managed by the resource provisioner unit. When a mobile client uses a cloud, the multimedia requests are sent to the data center, where the requests are stored in the RH queue. The requests are embedded with user context information as shown in Fig. 1.

In our proposed scheme, the client contexts include cell ID (*Cellid*), connection quality ( $C_T$ ), and mobile battery level ( $B_s$ ) that are used to make requests routing decision. We assume that the connection quality falls into any one of the two contexts: Context C1 represents Wi-fi connection, and context C2 represents the 3G/4G connection. We assume that the  $B_s$  falls into any one of the three values as shown in Fig. 1.

### 3.2 Problem Formulation

The VM requests are analyzed by RH based on the clients' context information and are labeled as high, medium, and low priority requests. Then, these requests are scheduled to the corresponding MS queue that maps the tasks to the resources while minimizing response time experienced by the client and the cost of running the resources in the cloud. This model has a single entry point RH which is represented by a M/M/1 queue, with an arrival and service rate modeled as an exponential random variables  $\lambda$  and  $\mu$ , respectively, in cloud computing where  $\lambda < \mu$ . The main purpose of RH is to schedule the requests to the MS based on the criticality of the client context information that is periodically sent by the client's context API. RH uses algorithm 1 for prioritizing the requests based on client context information. The service time of the RH queue is assumed to be exponentially distributed with mean service time  $\mu^{-1}$ , where  $\mu$  is the scheduling capacity of the RH.

The response time of the RH queue is given by  $T_{RH} = \frac{1/\mu}{1-\lambda/\mu}$ . Since there is no request loss in the previous system, the arrival of priority tagged requests at these three MS queue also follows Poisson process and the service time is assumed to be exponentially distributed with mean service time  $MS_i^{-1}$ . In this paper, we assume that, the possibility  $p_i$  of requests sent to each of the three  $MS_i$  is randomly generated. Thus, each of the three MS queue is modeled as an M/M/1 queuing system. According to the decomposition property of Poisson process, the possibility  $p_i$  of directing the request tasks with the priority  $i$  to the CS queue  $MS_i$ , where  $i = 1, 2, 3, \dots$ , impacts arrivals in each CS queue.

---

**Algorithm 1: Resource Handler Algorithm**

---

```

For Each new SRi(t)
  Assign_Priority SRi(t)
  If Priority (SRi(t))==1 then
    Push SRi(t)→ into MSHigh
  Else if Priority (SRi(t))==2 then
    Push SRi(t) → WL into MSMedium
  Else
    Push SRi(t) → WL into MSSmall
  End if
End for Each
Sub Assign_Priority SRi(t)
  If ((SRi → CT==C1)&&(SRi → Bs==Medium)) Then
    Priority (SRi(t)) = 2
  Else if ((SRi → CT==C2)&&(SRi→Bs==Low) || (SRi→Bs==Medium))
    Priority (SRi(t)) = 1
  Else
    Priority (SRi(t)) = 3
  End if
End Sub
    
```

Hence, the mean arrival rate is  $p_i\lambda$ . The response time at each computing server is  $T_{CS} = \sum_{i=1}^3 \frac{MS_i}{1-p_i\lambda/MS_i}$ . After processing the requests at MS, the service results are sent back to the customers. Average time a customer spends in the system (W) is defined as

$$W = T_{RH} + T_{CS} = \frac{1/\mu}{1-\lambda/\mu} + \frac{1/MS_i}{1-p_i\lambda/MS_i} \tag{1}$$

The mean server utilization is derived as  $\rho = \lambda e/\mu$ , where  $\lambda e$  is the mean arrival rate. We also drive  $P_0$  the probability that there are no customers in the system =  $1-\rho$

And  $P_n$ , the probability that there are n customers in the system is  $P_n = \rho^n P_0 P_{n+1}$ .

We assume that the load balanced resource allocation is achieved with the following  $P_0 < \rho < P_n$ . The total cost is calculated according to the utilized resources by the time. The resources include the resources at the request handler queue and media compute servers. The total cost is derived as

$$C = (\phi 1RH + \phi 2 \sum_{i=1}^n MS_i)t \tag{2}$$

where RH, MS is the service rate of the request handler and allocator at time t.

$\phi_1$  and  $\phi_2$  are the costs of RH and MS per request, respectively.

The transmission time is calculated as  $T = DR / \text{Task size}$  (3)

where DR is the data rate of the mobile device at the submission of job to the cloud

The turnaround time (TAT) is calculated as

$$TAT = W + T_{Mobile}^{Cloud} + T_{Cloud}^{Mobile} \quad (4)$$

where  $T_{Mobile}^{Cloud}$  and  $T_{Cloud}^{Mobile}$  are the transmission times from the mobile device to the cloud and the cloud output task to the mobile device, respectively.

---

### Algorithm 2: Cuckoo based Allocation Algorithm

---

Initialize Allocation Map

For Each Workload Queue

  While (MS<sub>(High/Medium/Small)</sub> != Empty)

    For Each new Virtual machine request,

      Extract Deadline, Initial arrival time, QoS

      Call CSA()

      Update Allocation map

      Access Code/ Image Repository

  End For Each

End While End For Each

Sub CSA (Fitness Function)

  Initialize population randomly, Choose best nest

  While (t < maxGeneration) or (stop criterion)

    Get a cuckoo randomly by Levy Flights

    Perform new nest

    The fitness function

$$\mathbf{F}(\mathbf{x}) = \mathbf{Max} \sum_1^n (\mathbf{x}\rho - \mathbf{y}W - \mathbf{z}C) \quad /* \text{As per Equation 5} */$$

    Evaluate its fitness/quality

    Choose a nest among n (say j) randomly

    If ( $F_i > F_j$ ) Replace j by the new solution End if

    A fraction (p) of worse nets are abandoned and new ones are built Keep the best solutions (or nests with quality solutions)

    Rank the solutions and find the current best

  End while

Post process results and visualization

The objective function of the optimization problem is as follows

$$F(x) = \text{Max} \sum_1^n (x\rho - yW - zC) \tag{5}$$

Subject to

$$TAT \leq \text{Deadline} \tag{6}$$

$$P_0 < \rho < P_n$$

$$\Psi(C_i(t)), \Psi(M_i(t)), \Psi(B_i(t)) \geq 0, \forall \text{ media server } i \text{ at time } t \tag{7}$$

$$\Psi(C_i(t)) \geq rC_j, \Psi(M_i(t)) \geq rM_j, \Psi(B_i(t)) \geq rB_j \tag{8}$$

$\forall \text{ Media server } i \text{ at time } t.$

where  $\Psi(C_i(t))$ ,  $\Psi(M_i(t))$ ,  $\Psi(B_i(t))$  are the percentage of the free capacity, memory, and bandwidth resources on the media server and  $rC_j$ ,  $rM_j$ ,  $rB_j$  are the resource requirements of the VM request  $i$  at time  $t$ .

### 3.3 Cuckoo-Based Resource Allocation Algorithm

Once the requests are scheduled to the corresponding MS, the optimal allocation of resources is done with cuckoo search-based allocation algorithm (CSA) as described in algorithm 2. CSA is a new meta-heuristic algorithm inspired by the obligate interspecific brood parasitism of some cuckoo species that lay their eggs in the nests of other host birds [2]. In this paper, we assume that a task reaching a MS pool may be processed locally and not be migrated to another MS. At each MS server pool, the CSA is applied to map the VM task to the suitable physical server. Here, we map cuckoo nests as the physical resources, cuckoo as the MS, and cuckoo's egg as the newly arrived task. This algorithm considers the following three rules

1. Each cuckoo lays one egg at a time and dumps it in a randomly chosen nests
2. The best nests with high quality of solutions will carry over to the next generations
3. The number of available nests are fixed, and a host can discover an alien egg with a probability  $p_a \in [0, 1]$

The generations of new solutions is done using Levy flights [2].



## 4 Performance Evaluation

This section presents simulation-based performance study of our proposed algorithm. In our simulation set up, there are two major simulation components: workload generator and cloud simulator. The workload generator is responsible for generating the workload, arbitrarily included mobile context information such as mobile energy level, connection quality, cell ID, time at which the context information is gathered. Google cloud traces from a cell of 12000 machines over about a month period in May 2011 was considered as the VM requests [12].

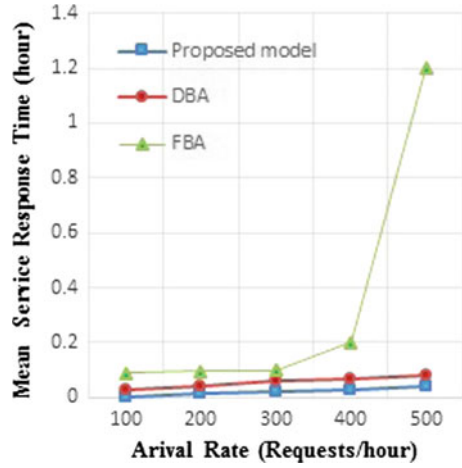
The simulation was done using the discrete event simulator Cloud Sim 3.0.3 executed on NetBeans. All two proposed algorithms have been implemented in the simulated environment. Table 1 shows the characteristics of the resources and the workload that have been used for all the experiments. We assume that the scheduling probability for the three MS servers is set as  $P = \{0.2, 0.3, 0.4\}$ . The RH server is charged by  $x = 0.12\$/\text{request}$ . The resource cost constraint is set to \$50.

We first compare the performance between the proposed adaptive resource provisioning scheme, in which the resources for the RH queue and MS queue are optimally allocated by solving the optimization problem and the FIFO-based equal resource allocation scheme, in which the requests are scheduled based on FIFO policy and the deadline-based allocation scheme. The comparison of the mean service response time between the proposed algorithm and other two state-of-the-art algorithms are shown in Fig. 2. In that, we can see that the proposed scheme achieves much lower response time compared to the other two. We next evaluate the percentage of deadline met, budget, and the server utilization level under different request arrival rate. The influences of change in no. of requests on the percentage of deadline met, resource cost, and the server utilizations are shown in Fig. 3, Fig. 4, and Fig. 5, respectively. The resource cost and percentage of

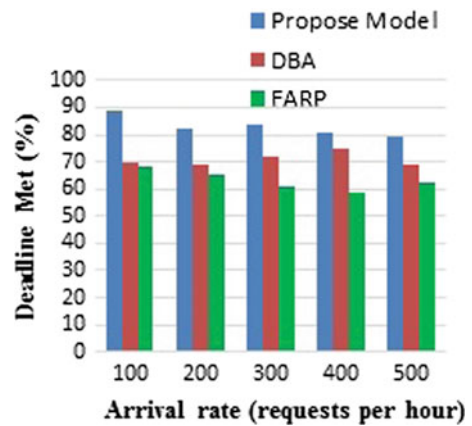
**Table 1** Simulation parameters

Parameters	Values		
	MS large	MS medium	MS small
Mean request arrival rate	500–600 requests/h		
Bandwidth (B/S)	3000	2000	1000
Size of workloads	15000 MB	10000 MB	7000 MB
No. of Pes per machine	4 (40000 MIPS)	3 (30000 MIP)	2 (1000–20000 MIPS)
Cost per workload	\$1-\$5		
Memory size (MB)	12,576	7168	2048
Cloud workload output size (MB)	300 + (10–50%)	300 + (10–50%)	300 + (10–50%)

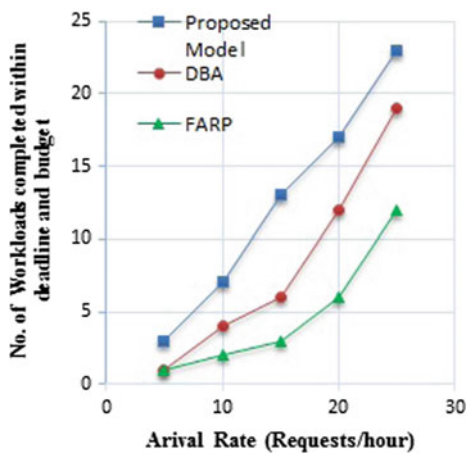
**Fig. 2** Average service response time versus arrival rate



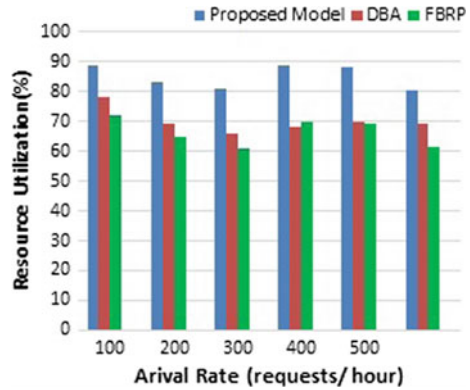
**Fig. 3** Comparison of the percentage of requests that met the deadline



**Fig. 4** Comparison of the customer satisfaction level



**Fig. 5** Resource utilization versus arrival rate



deadline violation by various workload tasks in ARPM is 30–40% lesser than the other two approaches where the critical mobile clients may severely facing the deadline violation. This time deviation is relatively substantial. The variation in average resource utilization is also noticeable.

## 5 Conclusion

This paper proposes an adaptive resource provisioning method for enhancing the quality of user experience. In addition to the regular workload parameters, the proposed model uses the client context information to solve the diverse mobile cloud parameters, which affects the cloud resource allocation. We model the system with queueing model to capture the relationship between service response time, cost-and the server utilization. The resource allocation problem is solved using cuckoo-based algorithm. The experiments conducted aim to evaluate and analyze the comparison of ARPM with other two algorithms. Further, this model can be improved to adapt the context changes.

## References

1. Dinh, H.T., Lee, C., et al.: A survey of mobile cloud computing: architecture, applications and approaches. In: Proceedings of Wireless Communications and Mobile Computing 13(18): 1587–1611 (2013)
2. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **1**, 330–343 (2010)
3. Garg, S.K., Gopalaiyengar, S.K., Buyya, R.: SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter. *IEEE ICA3PP 2011, Melbourne, Australia* (2011)

4. Buyya, R., Garg, S.K., et al.: SLA-oriented resource provisioning for cloud computing: challenges, architecture, and solutions. *Proc. IEEE Int. Conf. Cloud Serv Comput.* (2011)
5. Christian, V., et al.: Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka. *Elsevier J. Future Gener. Comput. Syst.* 58–65 (2012)
6. Calheiros, R.N., Vecchiola, C., et al.: The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid clouds. *Future Gener. Comput. Syst.* (2012)
7. Park, J., Kim, Y.S., Jeong, E.: Two-phase grouping-based resource management for big data processing in mobile cloud. *Int. J. Commun. Syst.* (2013)
8. Sood, S.K., et al.: Matrix based proactive resource provisioning in mobile cloud environment. *Elsevier J. Simul. Model. Pract. Theory* (2014)
9. Song, B., et al.: A two stage approach for task and resource management in multimedia cloud environment. *Springer Comput.* **98**, 119–145 (2016)
10. Durga, S., Mohan, S., et al.: Cuckoo based resource allocation for mobile cloud environments. *Comput. Intell. Cyber Secur. Comput. Models* **412**, 543–550 (2016)
11. Brendan, J, Rolf, S.: Resource management in clouds: survey and research challenges. *J. Netw. Syst. Manage.* 1–53 (2014)
12. Wilkes, J., Reiss, C.: Details of the ClusterData-2011-1 trace. [Online]. <https://code.google.com/p/>. (2011)

# Degree of Match-Based Hierarchical Clustering Technique for Efficient Service Discovery



P. Premalatha and S. Subasree

**Abstract** Clustering is an essential process in discovering services to fulfill the needs of the clients. There are several issues in clustering the services such as fulfilling the client's need, threshold computation and selecting an appropriate method of threshold calculation, and computing Inter-Cluster Distance (ICD). To resolve these issues, this paper proposes a novel Degree of Match-Hierarchical Clustering Technique (DoM-HCT). This technique utilizes Output Similarity Model (OSM) and Total Similarity Model (TSM) to make the clustering process more efficient. Extra levels are added to the TSM to improve the DoM. Only outputs are used by OSM, whereas both inputs and outputs are used by TSM. The incorrect clustering of services is avoided, and the demands are unaltered, while choosing threshold-ICD. The proposed DoM-HCT yields maximum precision and recall rate than the existing approaches.

**Keywords** Degree of Matching-Hierarchical Clustering Technique (DoM-HCT) Service discovery • Inter-Cluster Distance (ICD) • Threshold-ICD Output Similarity Model (OSM) and Total Similarity Model (TSM)

## 1 Introduction

Automatic discovery of services from various domains is essential for complex business requirements. Semantics-based revelation is tedious because of the utilization of semantic thinking which distinguishes semantic relations through various Degree of Match (DoM)s, in particular, correct, module, subsumes, and fail that may exist among questioned and accessible ideas. It is absolutely fundamental to

---

P. Premalatha (✉)  
Bharathiar University, Coimbatore, Tamilnadu, India  
e-mail: premalathap2000@gmail.com

S. Subasree  
Department of Computer Science & Engineering and Information Technology,  
Nehru College of Engineering and Research Center, Coimbatore, Tamilnadu, India

diminish the time taken for revelation as critical business forms include many administrations from various areas. Semantics-based closeness is promising for bringing precision, mechanization, and more dynamism into revelation. The primary part of this work is to present extra levels of DoMs while processing semantic similitude. Existing semantic methodologies [1, 2] figure the closeness between questioning idea and an accessible idea through four distinct levels of DoM, to be specific, correct, module, subsumes, and fizzle. In any case, these four levels recognize semantic relatedness of an idea just with regard to its super/subclasses, while extra levels of DoM are required to meet out the vast majority of the likeness requests of customers.

The remaining sections in the paper are structured as follows: Sect. 2 describes a short review of the existing clustering approaches. The proposed dam-HCT is explained in Sect. 3. Section 4 presents the performance analysis of the proposed DoM-HCT. The conclusion of the proposed work is discussed in Sect. 5.

## 2 Related Works

The ability of the Web service search engine is improved by grouping the services with similar functionalities through the clustering of Web services. Du et al. [3] presented a novel method for analyzing and substituting the Web services and proposed a service cluster unit. The performance analysis shows the efficiency of the proposed method when compared with the state-of-the-art methods. An efficient approach for the discovery of Web service using hierarchical clustering based on the distance measure is proposed [4]. The proposed approach achieved a significant improvement in the time complexity with the acceptable loss in the precision value. A non-logic-based matchmaking approach is presented for extracting the topic from the descriptions of semantic service and modeling the correlation between the extracted topics [5]. The service discovery can be achieved easily by using the concept lattice. Wu et al. [6] proposed a hybrid strategy for the recommendation of Web service. A novel approach is introduced by integrating tagging data and Web Service Description Language (WSDL) documents through augmented Latent Dirichlet Allocation (LDA) [7]. The efficiency of clustering the Web service is improved. The service similarity is combined with the frequency-inverse document frequency values of service names to identify the center of the cluster [8]. The natural language processing methods are used to develop the framework for discovering the semantic Web services [1]. A novel technique for clustering the service documents into service groups that are functionally similar to each other is proposed [2]. The Cat Swarm Optimization Algorithm is used for the clustering process. A novel ontology learning method is proposed by computing the semantic similarity of Web services [9]. The novel logic-based filters are defined for calculating the similarity of Web services. Tian et al. [10] suggested a new approach for the heterogeneous clustering of Web services based on the transfer learning from the extensive text data obtained from Wikipedia. A novel Dual Tag-aided

LDA (DT-LDA) is introduced to handle the variations in the semantics between service descriptions and auxiliary data.

### 3 DoM-HCT

The OSM is utilized to cluster the services that results in the same output. The outputs of the clusters are concatenated, and it is used as a label for that particular cluster. When an input query arrives, it is compared with the label of each cluster to get the exact match for that query. TSM is used to match the input and output of the query. In addition to OSM and TSM, the levels of DoM are used to find the similarity among the available services. To derive detailed information about the sub-clusters, a hierarchical clustering algorithm is applied. The strategies for discovering services in the proposed system are:

- The demands of clients are satisfied by the identification of semantic relations using the extra levels available in the DoM.
- OSM identifies similar services with respect to the output, whereas TSM identifies similar services according to both input and output.
- ICD is calculated from any of the three following methods such as single linkage, average linkage, and complete linkage.
- A novel method to choose the threshold-ICD is introduced.

Highly similar clusters are merged together by repeated checking of all the available services in each cluster. The techniques used for merging the clusters are single linkage, average linkage, and complete linkage, among which the complete linkage is best suited for merging services with large distance. The different levels of DoMs are utilized to set the threshold-ICD value based on the demands of clients. The value of threshold-ICD is assigned to 1, when the demand of the clients is fulfilled. Two services  $S_1$  and  $S_2$  from two different clusters  $C_1$  and  $C_2$  are merged based on the threshold value. The threshold-ICD is computed using the following equation:

$$\text{Threshold-ICD} = \frac{(m+n) \times \text{similarity\_demand}}{2mn} \quad (1)$$

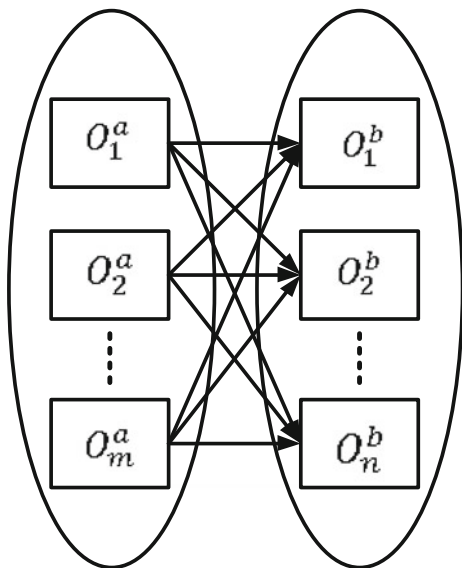
where  $m$  is the output parameters of service  $S_1$ , and  $n$  is the output parameters of service  $S_2$ . The relevant cluster is identified by matching the query along with every cluster using the TSM that is briefly explained in the following sections.

#### Output Similarity Model (OSM)

The OSM is described by merging services, namely A and B each containing a number of output parameters. The number of output parameters available in A and B is  $m$  and  $n$ , respectively. The output parameters of A are  $o_1^a, o_2^a, \dots, o_m^a$ , and the

output parameters of B are  $o_1^b, o_2^b, \dots, o_n^b$ . All the parameters in service A are matched with the parameters of service B based on semantic relation to calculate the similarity. Different levels of DoM are utilized to express the relationships of semantics. The levels of DoM are extended by the addition of four extra levels, such as plug-in, exact, fail, and subsumes. The degree of match between  $o_i^a$  and  $o_j^b$  is represented as  $\text{DoM}(o_i^a, o_j^b)$ , where  $o_i^a$  is the output parameter of service A, and  $o_j^b$  is the output parameter of service B. The matching between the cluster of two services is shown in Fig. 1. The  $\text{DoM}(o_i^a, o_j^b)$  is said to be exact if and only if  $o_i^a = o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be direct plug-in, when  $o_i^a$  is the direct superclass of  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be indirect, when  $o_i^a$  is the indirect superclass of  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be direct subsumes, when  $o_i^a$  is the direct subclass of  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be indirect subsumes, when  $o_i^a$  is the indirect subclass of  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be in a common parent/sibling relationship, when the parent of  $o_i^a$  is same as  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be in a partial parent relationship, when at least any one of the parents of  $o_i^a$  is similar to any one of the parents of  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be in a grandparent relationship, when any one of the grandparents of either  $o_i^a$  or  $o_j^b$  is similar to any one of the parents of either  $o_i^a$  or  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to have common child, when a single child has two parents, namely  $o_i^a$  and  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to have at least one common grandchild, when one grandchild has to grandparents, namely  $o_i^a$  and  $o_j^b$ . The  $\text{DoM}(o_i^a, o_j^b)$  is said to be fail, when no semantic relationships are satisfied.

**Fig. 1** Services and outputs





The pairs generated by  $o_1^a$  from the output parameters of A and B are  $(o_1^a, o_1^b)$ ,  $(o_1^a, o_2^b)$ , ...,  $(o_1^a, o_n^b)$ , and the pairs generated by  $o_2^a$  from the output parameters of A and B are  $(o_2^a, o_1^b)$ ,  $(o_2^a, o_2^b)$ , ...,  $(o_2^a, o_n^b)$ . Finally, the following pairs such as  $(o_m^a, o_1^b)$ ,  $(o_m^a, o_2^b)$ , ...,  $(o_m^a, o_n^b)$  are generated by  $o_m^a$ . The output similarity of A and B can be expressed as

$$\begin{aligned} \text{outsim}(A, B) = \frac{1}{m} \times & \left[ \max \left( \text{DoM}(o_1^a, o_1^b), \text{DoM}(o_1^a, o_2^b), \dots, \text{DoM}(o_1^a, o_n^b) \right) \right. \\ & + \max(\text{DoM}(o_2^a, o_1^b), \text{DoM}(o_2^a, o_2^b), \dots, \text{DoM}(o_2^a, o_n^b)) \\ & + \dots \max(\text{DoM}(o_m^a, o_1^b), \text{DoM}(o_m^a, o_2^b), \dots, \text{DoM}(o_m^a, o_n^b)) \left. \right] \end{aligned} \quad (2)$$

The output similarity of B and A is defined as

$$\begin{aligned} \text{outsim}(B, A) = \frac{1}{m} \times & \left[ \max \left( \text{DoM}(o_1^b, o_1^a), \text{DoM}(o_1^b, o_2^a), \dots, \text{DoM}(o_1^b, o_m^a) \right) \right. \\ & + \max(\text{DoM}(o_2^b, o_1^a), \text{DoM}(o_2^b, o_2^a), \dots, \text{DoM}(o_2^b, o_m^a)) \\ & + \dots \max(\text{DoM}(o_n^b, o_1^a), \text{DoM}(o_n^b, o_2^a), \dots, \text{DoM}(o_n^b, o_m^a)) \left. \right] \end{aligned} \quad (3)$$

$$\text{Sim}(A, B)_{OSM} = \frac{1}{2} \times (\text{OutSim}(A, B) + (\text{OutSim}(B, A))) \quad (4)$$

### Total Similarity Model

The total similarity of A and B is given as

$$\text{Sim}(A, B)_{TSM} = 0.5 \times (\text{Sim}(A, B)_{OSM} + \text{InputSim}(A, B)) \quad (5)$$

where  $\text{Sim}(A, B)_{TSM}$  is termed as the normalized output similarity, and  $\text{InputSim}(A, B)$  is the normalized input similarity. The input similarity can be determined using the following equation

$$\text{InputSim}(A, B) = 0.5 \times \text{InSim}(A, B) + \text{InSim}(B, A) \quad (6)$$

### Clustering of Services using OSM

The similarity score of all the service pairs available in the repositories is determined using OSM. The dissimilarity of the computed similarity scores is calculated as

$$Dissim(s_1, s_2) = (1 - Sim(s_1, s_2)) \quad (7)$$

The hierarchical clustering algorithm takes the  $N \times N$  dissimilarity matrix as input. At the initial stage, a single service is assigned to each cluster, then, highly matching clusters are combined together as a single cluster. The degree of match of the newly merged cluster is compared with the other existing clusters. These steps are repeated until; the threshold-ICD gets satisfied.

The similarity score of two clusters  $C_1$  and  $C_2$  computed using the single linkage and complete linkage method is

$$Sim(C_1, C_2) = \max\{sim(a, b): a \in C_1, b \in C_2\} \quad (8)$$

where the services of  $C_1$  and  $C_2$  are denoted by  $a$  and  $b$ , respectively. The similarity score computed using the average linkage method for two different clusters  $C_1$  and  $C_2$  is as follows

$$Sim(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{a \in C_1} \sum_{b \in C_2} sim(a, b) \quad (9)$$

where  $|C_1|$  and  $|C_2|$  are the cardinality of  $C_1$  and  $C_2$ , respectively.

## 4 Performance Analysis

The prime goal is to utilize OSM for clustering services by analyzing the single linkage, complete linkage, and average linkage methods to find the better clustering solution that satisfies the threshold-ICD. The next goal is to analyze the level of performance optimization via clustering. The query that discovers the services in a minimum computation time is referred to as having better performance. OSM is utilized to calculate the pair of services available in the test data, and this experiment is implemented using Java. Jena API and Pellet reasoned are used to determine various levels of DoM with the help of OSM. The clustering tool obtains dissimilarity matrix as input, and the representation module is used to label the output clusters. The cluster details and their corresponding labels are stored for easy retrieval of clusters on the submission of a query. Then, TSM is used to identify similar queries from the highly similar clusters. The experimental setup is illustrated in Fig. 2. Figure 3 shows the r-p precision curve. The proposed DoM-HCT is compared with the Web Service Operation Discovery algorithm (OpD), OpD&Single, ServicePool, LinkedData, Schema Matching, and Keyword-based matching [11]. The OpD can discover the composite Web services, and OpD&Single can discover the single Web services only. This results in the reduction in the precision rate of OpD&Single. The r-p curve for the OpD&Single drops significantly with the increase in the number of discovery results.

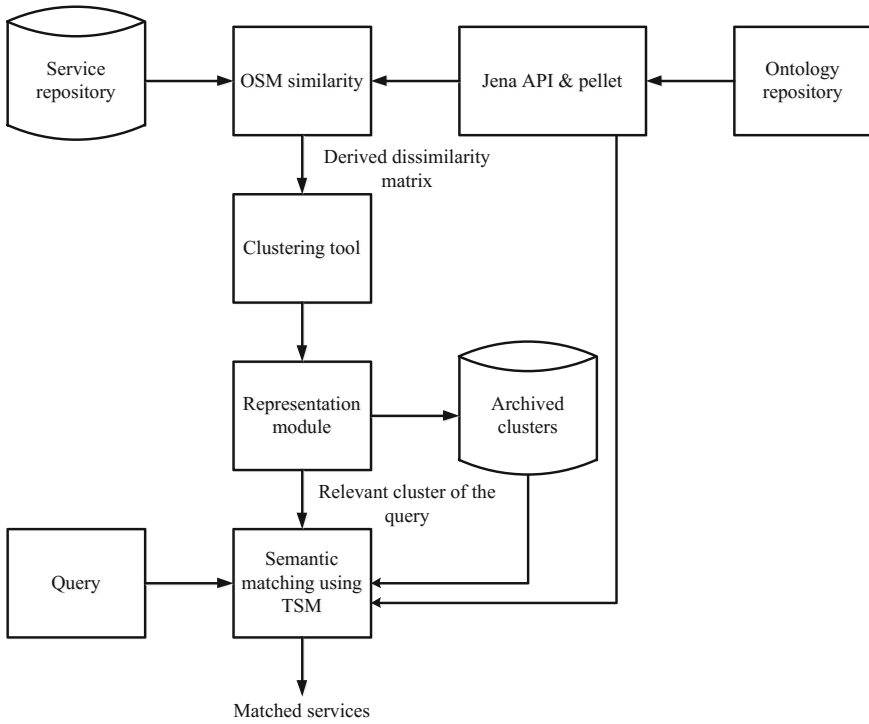


Fig. 2 Experimental setup

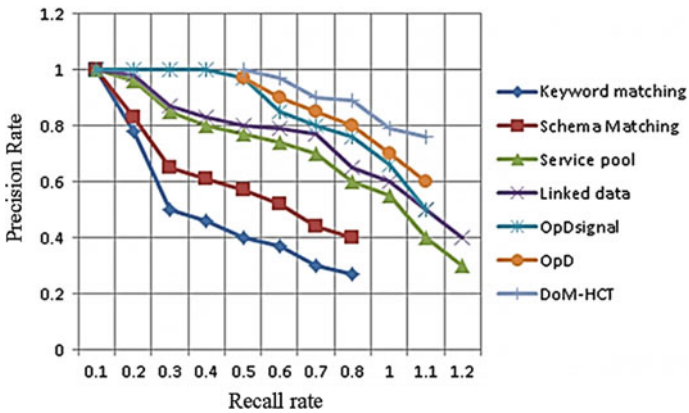


Fig. 3 r-p precision curve

The OpD and OpD&Single perform mining of the combinations of fragmented terms with high probability, and the ServicePool can mine the terms having high frequencies in same domains. The ServicePool cannot offset the drawbacks disadvantages in mining semantics. Hence, the precision rate of the ServicePool method is lower than the OpD and OpD&Single. As the schema-matching approach considers the XML structure, it outperforms the keyword-based matching approach. The precision rate of the keyword-based and schema-matching approaches is low, as it ignores the underlying semantics. The schema-matching and keyword-based matching approaches are purely based on string matching without any semantic extensions. The recall rate of these methods are lower than the OpD&Single. The proposed DoM-HCT method combined the OSM and TSM models to meet the query requirements. The highly matching services are grouped as a single cluster to make the search process easier. Hence, the proposed method yields maximum precision rate and recall rate than the existing approaches.

## 5 Conclusion

In this work, DoM-HCT clustering method is proposed for service discovery to satisfy the demands of the customers. This method combined OSM and TSM models to meet the query requirements. The highly matching services are grouped as a single cluster to make the search process easier. The outputs are considered by OSM, and both the input and outputs are considered by TSM for grouping services. The threshold-ICD is computed using three methods simple, complete, and average linkage. From the analysis, it is concluded that the complete linkage method is more efficient in the calculation of threshold-ICD. From the comparative analysis, it is concluded that the proposed DoM-HCT method achieves better precision and recall rate than the OpD, OpD&Single, ServicePool, LinkedData, Schema Matching, and Keyword-based matching.

## References

1. Sangers, J., Frasinca, F., Hogenboom, F., Chepegin, V.: Semantic web service discovery using natural language processing techniques. *Expert Syst. Appl.* **40**, 4660–4671 (2013)
2. Kotekar, S., Kamath, S.S.: Enhancing service discovery using cat swarm optimisation based web service clustering. *Perspect. Sci.* **8**, 715–717 (2016)
3. Du, Y., Gai, J., Zhou, M.: A web service substitution method based on service cluster nets. *Enterp. Inf. Syst.* 1–17 (2016)
4. Cong, Z., Fernandez, A., Billhardt, H., Lujak, M.: Service discovery acceleration with hierarchical clustering. *Inf. Syst. Front.* **17**, 799–808 (2015)
5. Aznag, M., Quafafou, M., Jarir, Z.: Leveraging formal concept analysis with topic correlation for service clustering and discovery. In: *IEEE International Conference on Web Services (ICWS)*, pp. 153–160 (2014)

6. Wu, J., Chen, L., Zheng, Z., Lyu, M.R., Wu, Z.: Clustering web services to facilitate service discovery. *Knowl. Inf. Syst.* **38**, 207–229 (2014)
7. Chen, X., Zheng, Z., Liu, X., Huang, Z., Sun, H.: Personalized qos-aware web service recommendation and visualization. *IEEE Trans. Serv. Comput.* **6**, 35–47 (2013)
8. Kumara, B.T., Paik, I., Chen, W., Ryu, K.H.: Web service clustering using a hybrid term-similarity measure with ontology learning. *Int. J. Web Serv. Res. (IJWSR)* **11**, 24–45 (2014)
9. Rupasingha, R.A., Paik, I., Kumara, B.T., Siriweera, T.A.S.: Domain-aware web service clustering based on ontology generation by text mining. In: *IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–7 (2016)
10. Tian, G., Sun, C., He, K.-Q., Ji, X.-M.: Transferring auxiliary knowledge to enhance heterogeneous web service clustering. *Int. J. High Perform. Comput. Netw.* **9**, 160–169 (2016)
11. Cheng, B., Li, C., Chen, J.: A web services discovery approach based on interface underlying semantics mining. In: *IEEE Transactions on Knowledge and Data Engineering* (2016)

# Providing Confidentiality for Medical Image—An Enhanced Chaotic Encryption Approach



M. Y. Mohamed Parvees, J. Abdul Samath and B. Parameswaran Bose

**Abstract** This study presents an encryption algorithm to secure the medical images using enhanced chaotic economic map. The different enhanced chaotic economic maps are derived and studied with respect to their bifurcate nature and Lyapunov exponents. The enhanced maps are utilized for chaotic sequence generations. These sequences are employed for confusing, diffusing, and swapping the 16-bit DICOM image's pixels, thereby assure confidentiality. After scrambling, the different security analyses such as statistical, entropy, differential, key space analysis are performed to prove the effectiveness of the proposed algorithm.

**Keywords** Patient confidentiality • Chaotic map • DICOM encryption

## 1 Introduction

The cloud and PACS services grow rapidly in the field of teleradiology along with the advent use of radiological information system. Hence, the patient confidentiality has become a cumbersome factor in a shared environment. In a cloud or virtual PACS, the patient confidentiality can be attained through sharing the data in encrypted form. The cloud service provider should provide an encryption technique to secure the patient's data. A very few encryption schemes for protecting patient

---

B. Parameswaran Bose—He was with Fat Pipe Network Pvt. Ltd., Mettukuppam, Chennai-600009, India and now as an Independent Researcher, #35, I Main, Indiragandhi Street, Udayanagar, Bangalore, India.

---

M. Y. Mohamed Parvees (✉)  
Research and Development Centre, Bharathiar University, Coimbatore 641046, India  
e-mail: yparvees@gmail.com

J. Abdul Samath  
Department of Computer Science, Government Arts College, Udumalpet 642126, India

B. Parameswaran Bose  
Fat Pipe Network Pvt. Ltd., Mettukuppam, Chennai 600009, India

data are proposed in imaging informatics that includes chaos-based encryption also [1–5]. Yet, DICOM security is an open research area which needs to be expedited to meet out challenges in DICOM sharing. In chaos-based encryption technique, the chaotic sequence generation is playing a pivotal role in accomplishing efficient encryption, thereby preventing various kinds of security attacks [6]. In order to provide effective chaotic encryption, the existing chaotic maps could be altered to achieve higher chaotic behavior [7]. In this regard, chaotic economic map could be enhanced to achieve higher chaotic behavior than the actual in terms of its nature of bifurcation and positive Lyapunov exponents. Though Parvees et al. [7] enhances the CEM and encrypts the medical image, the further three different new types of enhanced chaotic economic maps are derived and used for generating various permutation, masking, and swapping sequences to scramble the DICOM pixels.

## 2 Mathematical Background

The equation  $x_{n+1} = x_n + k \times [a - c - b \times (1 + \gamma) \times x_n^\gamma]$  represents the CEM.

Where  $a > 0$  is size of market demand,  $b > 0$  is slope of market price,  $c \geq 0$  is fixed marginal cost,  $\gamma$  is a constant ( $\gamma = 3$ ), and  $k > 0$  is speed of adjustment parameter.  $x_0$  is initial parameter lies between  $(0, 1)$ . The bifurcate range of CEM notifies that the chaotic behavior lies between  $(0, 0.35)$ . These studies use the ECEM type-1 which is reported by [7] and initiates the other three types of enhanced CEMs. The CEM is modified by multiplying the initial parameter  $x_n$  with sin and cos trigonometric functions as given in Eqs. (1–4). The ECEM type-1, type-2, type-3, and type-4 are given in Eqs. (1–4), respectively.

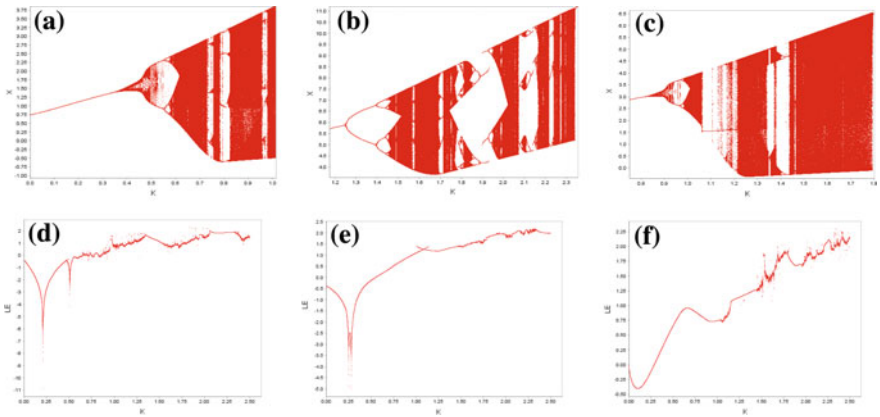
$$x_{n+1} = \sin x_n + k \times [a - c - b \times (1 + \gamma) \times (\cos x_n)^\gamma] \quad (1)$$

$$x_{n+1} = \cos x_n + k \times [a - c - b \times (1 + \gamma) \times (\cos x_n)^\gamma] \quad (2)$$

$$x_{n+1} = \cos x_n + k \times [a - c - b \times (1 + \gamma) \times (\sin x_n)^\gamma] \quad (3)$$

$$x_{n+1} = \sin x_n + k \times [a - c - b \times (1 + \gamma) \times (\sin x_n)^\gamma] \quad (4)$$

The bifurcate range and Lyapunov exponent values are illustrated for ECEM type-2, type-3, and type-4 and behave chaotic while  $k$  is in between  $(0.63, 1)$ ,  $(1.51, 2.35)$ , and  $(1.07, 1.80)$ , respectively. The ECEM exhibits higher bifurcate range and positive Lyapunov exponents than the CEM (Fig. 1a–f).



**Fig. 1** Bifurcate and Lyapunov exponents (LE) diagram of CEM and ECEM with respect to the control parameter ( $k$ ): **a** bifurcation of ECEM type-2, **b** LE of ECEM type-2, **c** bifurcation of ECEM type-3, **d** LE of ECEM type-3, **e** bifurcation of ECEM type-4, **f** LE of ECEM type-4

### 3 Methodology

The three processes, namely permutation, diffusion, and swapping, are employed in encrypting the DICOM pixel information. The four enhanced chaotic maps are iterated to generate the different kinds of sequences in which different maps use different input parameters that make the proposed system stronger and complex. These input parameters are encrypting keys which could be used during the decryption process also.

#### 3.1 Generation of Permutation Sequence

ECEM type-1 [7] generates the 64-bit double-valued chaotic sequences. These sequences are sorted in ascending or descending order to obtain the sorted sequences. The permutation sequence is obtained from the two sorted sequences. The pseudo-code for generating permutation sequence is given in pseudo-code 1. The proposed encryption algorithm uses the eight permutation sequences to scramble the DICOM pixel positions.



---

**Pseudo-code 1: Generating permutation sequence**


---

BEGIN

Chaotic sequence  $C = \{c_1, c_2, c_3, \dots, c_n\} \leftarrow SeqGeneration(a, b, c, \gamma, k)$  from chaotic Eq. (2).

Choose chaotic sequence  $C = \{c_{1000}, c_{1001}, c_{1002}, \dots, c_n\}$ .

$S = \{s_1, s_2, s_3, \dots, s_n\} \leftarrow Sort(C)$  by sorting  $C$ .

Let  $c_{1000} = a \cdot a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16}$

Calculate  $sum = ((100 - a_1 a_2) \times (100 - a_9 a_{10})) - ((100 - a_5 a_6) * (100 - a_{13} a_{14}))$

If  $sum = 0$ , group chaotic elements into one dimensional array  $d_0$

Repeat to get  $d_1, d_2, d_3, \dots, d_{10000}$  while  $sum = 1, 2, 3, \dots, 10000$  for remaining chaotic sequences.

Permutation sequence  $P = \{p_1, p_2, p_3, \dots, p_n\}$  is obtained by indexing  $\{d_0, d_1, d_2, \dots, d_{10000}\}$  with the elements of  $S = \{s_1, s_2, s_3, \dots, s_n\}$ .

END

### 3.2 Generation of Diffusion Sequence

The ECEM type-2 generates the 64-bit double-valued chaotic sequence. Since the 16-bit-valued DICOM pixels have the values between 0 and 65535, the chaotic sequences are converted to integer-valued sequence whose values lies between 0 and 65535. Then, the eight different diffusion sequences are employed to mask the DICOM pixels by altering the pixel values. The pseudo-codes for generating masking sequences are shown in pseudo-code 2.

---

**Pseudo-code 2: Generating masking sequence**


---

BEGIN

Chaotic sequence  $C = \{c_1, c_2, c_3, \dots, c_n\} \leftarrow SeqGeneration(a, b, c, \gamma, k)$  from chaotic Eq. (3).

Choose chaotic sequence  $X = \{x_{1000}, x_{1001}, x_{1002}, \dots, x_n\}$  from  $C$ .

Obtain masking sequence  $M = \{m_1, m_2, m_3, \dots, m_n\}$  from  $X = \{x_1, x_2, x_3, \dots, x_n\}$  by calculating

$M_i = \text{int} \{ [abs(x_i) - [abs(x_i)] \times 10^{16}] \text{ mod } 65535 \}$  where  $M_i \in (0, 65535)$ .

END

### 3.3 Generation of Swapping Sequence

The ECEM type-3 and type-4 are useful in generating the swapping sequences. The 64-bit double-valued sequences are obtained by iterating the ECEM type-3 and type-4 maps. Further, the integer-valued swapping sequences are obtained from the 64-bit two double-valued sequences. These swapping sequences are helpful in exchanging the DICOM pixel values. The sixteen different swapping sequences are generated using the pseudo-code 3.

---

Pseudo-code 3: Generating swapping sequence

---

BEGIN

Chaotic sequence  $C = \{c_1, c_2, c_3, \dots, c_n\} \leftarrow SeqGeneration(a, b, c, \gamma, k)$  from chaotic Eq. (2).

Choose chaotic sequence  $C = \{c_{1000}, c_{1001}, c_{1002}, \dots, c_n\}$ .

$S = \{s_1, s_2, s_3, \dots, s_n\} \leftarrow Sort(C)$  By sorting C.

Let  $c_{1000} = a \cdot a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16}$

Calculate  $sum = ((10 - a_1 a_2) \times (10 - a_9 a_{10})) - ((10 - a_5 a_6) \times (10 - a_{13} a_{14}))$

If  $sum = 0$ , group sequences into one dimensional array  $d_0$

Repeat to get  $d_1, d_2, d_3, \dots, d_{100}$  while  $sum = 1, 2, 3, \dots, 10000$  for remaining chaotic sequences.

Swapping sequence  $W = \{w_1, w_2, w_3, \dots, w_n\}$  is obtained by indexing  $\{d_0, d_1, d_2, \dots, d_{100}\}$  with the elements of  $S = \{s_1, s_2, s_3, \dots, s_n\}$ .

---

END

### 3.4 The DICOM Image Encryption Algorithm

The overall algorithm is given in pseudo-code 4 which is more complex, thereby becomes difficult to guess the run. The significance of the proposed algorithm is initiating swapping process in between the permutation and masking processes. At the end of the eighth iteration, the DICOM pixels are encrypted completely and become more random. The decryption is the reverse process of encryption.

---

Pseudo-code 4: The DICOM pixel image encryption algorithm

---

BEGIN

READ DICOM image

SET  $l \leftarrow image\ width$ , SET  $m \leftarrow image\ height$

CALCULATE  $n = l \times m$

READ 16-bit gray values in one dimensional array  $R = \{r_1, r_2, r_3, \dots, r_n\}$ .

CREATE 8 permutation ( $P_1, P_2, \dots, P_8$ ), 8 masking ( $M_1, M_2, \dots, M_8$ ) and 16 swapping ( $W_1, W_2, W_3, \dots, W_{16}$ ) sequences of length n using different ECEMs.

SET  $i \leftarrow 1$ , SET  $j \leftarrow 1$ , SET  $k \leftarrow 2$

for  $n \leftarrow 1$  to 8 do

  COMPUTE permuted pixel  $R_i$

  COMPUTE swapped pixel  $R \leftarrow R \Leftrightarrow W_j$

  COMPUTE diffused pixels  $R \leftarrow R \oplus M_i$

  COMPUTE swapped pixel  $R \leftarrow R \Leftrightarrow W_k$

  SET  $i \leftarrow i + 1$ , SET  $j \leftarrow j + 2$ , SET  $k \leftarrow k + 2$

---

end for

RETURN Encrypted pixel R.

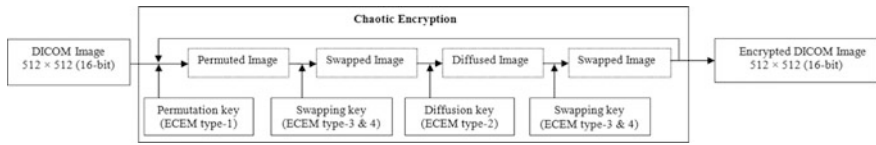


Fig. 2 Architecture of the DICOM image encryption

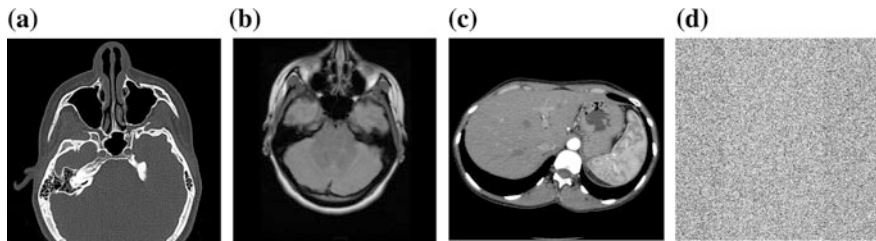


Fig. 3 Test images a MRI-1, b MRI-2, c CT-1, d encrypted MRI-1

Table 1 Results of correlation coefficient analysis of plain and cipher image

Test images	Image	Vertical correlation		Horizontal correlation		Diagonal correlation	
		Proposed	Ref. [2]	Proposed	Ref. [2]	Proposed	Ref. [2]
MRI-1	Plain	0.96117	NA	0.97315	NA	0.94088	NA
	Cipher	0.00440	NA	-0.00227	NA	0.00414	NA
MRI-2	Plain	0.97560	0.9756	0.96639	0.9661	0.94940	0.9469
	Cipher	-0.00604	-0.0060	-0.00354	-0.0060	0.00068	0.0011
CT-1	Plain	0.99487	0.9955	0.99658	0.9966	0.99191	0.9921
	Cipher	-0.00445	-0.0178	-0.00411	-0.1987	0.00719	0.0006

### 4 Experimental Results and Discussion

The three 16-bit DICOM images of size  $512 \times 512$  are considered to analyze the proposed algorithm (Fig. 2). The DICOM images are shown in Fig. 3a–d. The results of the various security analyses are shown in Tables 1 and 2.

The proposed DICOM encryption scheme is evaluated with various metrics to defy different kinds of attacks and compared with results of existing literatures [2, 4, 8–10]. The plain and cipher image sizes are same during the encryption and decryption processes. The results of the correlation coefficient, mean square error (MSE), peak signal noise ratio (PSNR), and mean-variance gray value analysis prove that the algorithm resists statistical attacks which are far better than existing algorithms [2–4, 7, 8, 10]. The algorithm resists differential attack based on number of pixel change rate (NPCR) and unified average change intensity (UACI) results

**Table 2** Results of mean-variance gray value, entropy, NPCR, UACI, MSE, and PSNR values

Test Images	Image	Mean-variance gray value		Entropy		NPCR		UACI		MSE		PSNR	
		Proposed	Ref [7]	Proposed	Ref [7]	Proposed	Ref. [2]	Proposed	Ref. [2]	Proposed	Ref. [2]	Proposed	Ref. [2]
MRI-1	Plain	575.53	7.08823	9.9152	NA	99.9980	NA	50.0877	NA	18949.29	47.5328		
	Cipher	16397.24	15.17716	15.8093	99.9984	99.9984	99.9971	49.9773	33.373	18904.99	47.5429		
MRI-2	Plain	184.39	10.44122	7.0882	15.80773	99.9984	99.9983	50.0266	33.396	18926.91	47.5379		
	Cipher	16375.85	15.1722	15.1722	NA	99.9984	99.9983	50.0266	33.396	18926.91	47.5379		
CT-1	Plain	503.51	7.0685	7.0685	NA	99.9984	99.9983	50.0266	33.396	18926.91	47.5379		
	Cipher	16384.28	15.8067	15.8067	NA	99.9984	99.9983	50.0266	33.396	18926.91	47.5379		

which are optimal [2–4, 7, 8]. The entropy values remain intact to resist entropy attacks that are similar to the values given in [4, 7, 8]. Since the four different enhanced chaotic maps are involved in this study, the key space is  $10^{3072}$  which is significantly higher than [7]. The total number of parameter keys used for the encryption process is 192. Each parameter key will be of 64-bit key length. Moreover, the confusion, diffusion, and substitution operations are executed for four rounds. Hence, the complexity of the algorithm is  $2^{64 \times 48} \times 2^{64 \times 48} \times 2^{64 \times 48} \times 2^{64 \times 48} \times 4$ . The proposed encryption technique can be included along with existing imaging informatics.

## 5 Conclusion

The four different enhanced maps are employed to permute, diffuse, and swap the DICOM pixels. Thus, the proposed cryptosystem is highly efficient and complex that could be sufficient to provide confidentiality for medical images. Further, the key space is very larger, thereby resists brute force attacks. The results of various security analyses and comparison with existing literatures are also proved that the system has high security and can be used in teleradiology without altering the existing architecture.

## References

1. Kobayashi, L.O.M., Fururie, S.S.: Proposal for DICOM multiframe medical image integrity and authenticity. *J. Digit. Imag.* **22**(1), 71–83 (2009). <https://doi.org/10.1007/s10278-008-9103-6>
2. Ravichandran, D., Praveenkumar, P., Rayappan, J.B.B., Amirtharajan, R.: Chaos based crossover and mutation for securing DICOM image. *Comput. Biol. Med.* **72**, 170–184 (2016). <https://doi.org/10.1016/j.combiomed.2016.03.020>
3. Praveenkumar, P., Amirtharajan, R., Thenmozhi, K., Rayappan, J.B.B.: Triple chaotic image scrambling on RGB—a random image encryption approach. *Secur. Commun. Netw.* **8**(18), 3335–3345 (2015). <https://doi.org/10.1002/sec.1257>
4. Fu, C., Meng, W., Zhan, Y., Zhu, Z., Lau, F.C.M., Tse, C.K., Ma, H.: An efficient and secure medical image protection scheme based on chaotic maps. *Comput. Biol. Med.* **43**, 1000–1010 (2013). <https://doi.org/10.1016/j.combiomed.2013.05.005>
5. Kanso, A., Ghebleh, M.: An efficient and robust image encryption scheme for medical applications. *Commun. Nonlinear Sci. Numer. Simul.* **24**(1–3), 98–116 (2015). <https://doi.org/10.1016/j.cnsns.2014.12.005>
6. Yavuz, E., Yazici, R., Kasapbas, M.C., Yamac, E.: A chaos-based image encryption algorithm with simple logical functions. *Comput. Electr. Eng.* **54**, 471–483 (2016). <https://doi.org/10.1016/j.compeleceng.2015.11.008>
7. Parvees, M.Y.M., Samath, J.A., Bose, B.P.: Secured medical images—a chaotic pixel scrambling approach. *J. Med. Syst.* **40**, 232 (2016). <https://doi.org/10.1007/s10916-016-0611-5>

8. Praveenkumar, P., Amirtharajan, R., Thenmozhi, K.: Medical data sheet in safe havens—a tri-layer cryptic solution. *Comput. Biol. Med.* **62**, 264–276 (2015). <https://doi.org/10.1016/j.combiomed.2015.04.031>
9. Dzwonkowski, M., Papaj, M., Rykaczewski, R.: A new quaternion-based encryption method for DICOM images. *IEEE T. Image Process.* (2015). <https://doi.org/10.1109/TIP.2015.2467317>
10. Zhang, S., Gao, T., Gao, L.: A novel encryption frame for medical image with watermark based on hyperchaotic system. *Math. Probl. Eng.* (2014). <https://doi.org/10.1155/2014/240749>

# A Novel Node Collusion Method for Isolating Sinkhole Nodes in Mobile Ad Hoc Cloud



Immanuel Johnraja Jebadurai, Elijah Blessing Rajsingh  
and Getzi Jeba Leelipushpam Paulraj

**Abstract** Cloud computing combined with mobile technology provides immense computing capabilities. The services and applications can be accessed anywhere and anytime. The required services for mobile users are offered on runtime with high reliability and availability by the mobile ad hoc cloud. Mobile devices communicate with each other through multi-hop communication using routing protocols. Heterogeneity of the devices, limited battery life, and mobility of the nodes are the salient features of the mobile devices. These characteristics impose greater challenges in terms of routing, security, and privacy. Any attacker node can advertise false routing information to lure other nodes to use its service. Such nodes are called sinkhole nodes. Sinkhole nodes need to be isolated from the mobile cloud as early as possible with high precision to provide uninterrupted service to other mobile users. This paper proposes a node collusion method for the detection and isolation of sinkhole nodes. The proposed method has been implemented using NS-2 simulator. The results were obtained. The results were compared with the existing state-of-the-art algorithms for sinkhole detection. It is found that the proposed method outperforms other methods in terms of detection time, false-positive ratio, routing overhead, and packet delivery ratio.

**Keywords** Mobile cloud · Cloud computing · Sinkhole attack  
Collusion · RREQ · RREP · DSR · Security

---

I. J. Jebadurai (✉) · E. B. Rajsingh · G. J. L. Paulraj  
Karunya University, Coimbatore, India  
e-mail: immanueljohnraja@gmail.com

E. B. Rajsingh  
e-mail: elijahblessing@gmail.com

G. J. L. Paulraj  
e-mail: getzi@karunya.edu

## 1 Introduction

Cloud computing provides platform, infrastructure, and software as services on demand. Mobile technology has revolutionized the world due to its anywhere anytime service [1]. Smarter applications, reduced cost, and high inbuilt computing power have increased the use of mobile users. However, many online applications such as video, audio, and image processing are compute-intensive. Cloud provides compute and storage as a service for such mobile applications on demand. By mobile cloud computing technology, high-intensive compute powers are made available for the mobile users. This demands mobile users to be connected with the cloud every time the application needs computing resource [2, 3]. This requirement imposes challenges in terms of scheduling, task migration, and quality of service.

In contrast, mobile ad hoc cloud uses computing resource from nearby nodes for online applications. They communicate among themselves to form ad hoc environment to provide and receive service from and to its peer. The mobile ad hoc cloud is formed using traditional ad hoc routing protocols such as dynamic source routing (DSR) protocol and ad hoc on-demand distance vector (AODV) protocol. Mobile ad hoc cloud is useful in various applications such as disaster recovery, entertainment, emergency services, home and enterprises, and tactical networks. The reliability of the mobile devices is affected by transmission impediments. The limited radio band exhibits less data rate when compared with wired networks. Mobile ad hoc cloud experiences heavy packet loss due to hidden and exposed terminal problems. The dynamic characteristics of the topology result in frequent network disjoint. There is no central coordination in mobile cloud. This limitation prevents the usage of firewall. Spoofing attack, eavesdropping, and denial of service are few of the other attacks on mobile cloud.

Sinkhole attack is one among the prime security challenges in mobile cloud. Sinkhole attackers attempt to lure all the traffic in the mobile cloud toward itself, thereby bringing down the performance of the cloud. Sinkhole attack also expedites snooping, packet drop, denial of service, and modification of packet. This compromises the quality of service for the mobile users. Hence, sinkhole nodes need to be detected and isolated from the mobile cloud network as soon as possible with high precision.

This paper proposes a node collusion method to detect and isolate sinkhole attack in the mobile ad hoc cloud. The mobile nodes collaborate among themselves to detect, and they isolate the node by exchanging target alert messages. The proposed method has been implemented and experimented using NS-2 simulator. The experimental results are evaluated in terms of detection time, false-positive rate, packet delivery ratio, and routing overhead. Section 2 discusses the mobile ad hoc cloud architecture and sinkhole attack. Section 3 reviews various literature available for the detection and isolation of sinkhole nodes. Section 4 gives the proposed node collusion detection and isolation method. Implementation aspects are given in Sect. 5. It also covers evaluation results. Conclusions are provided in Sect. 6 with future research directions.



## 2 Mobile Ad Hoc Cloud Architecture

The combined capabilities of mobile computing and cloud computing powers the mobile ad hoc cloud [4–6]. The formation of mobile ad hoc cloud requires routing protocols. Routing protocols are classified as proactive and reactive. Proactive routing tries to maintain up-to-date information about the entire network. Reactive routing protocol works on demand. The route is formed whenever required. Mobile ad hoc cloud is formed using reactive routing protocol because the network is formed whenever the service requirement arises. The examples of reactive routing protocol include AODV, DSR, and TORA.

### Sinkhole Attack

A sinkhole node attack by malicious node S is depicted in Fig. 1. Here, malicious node S generates fake route request packet as if it is created from node F.

Every node receiving the route request updates its routing cache that node S is one-hop distance from F. This message is updated as it holds higher sequence number. This way, the malicious node enters the routing cache of every node and captures the traffic. Such sinkhole attack is a serious threat in the mobile ad hoc cloud. It facilitates other passive and active attacks.

## 3 Related Works

The existing methodologies in the literature are classified into network monitoring-based detection, detection methods by modification of protocols, agent-based detection, anomaly detection, intrusion detection systems (IDS), detection methods using cryptographic techniques, trust-/reputation-based detection

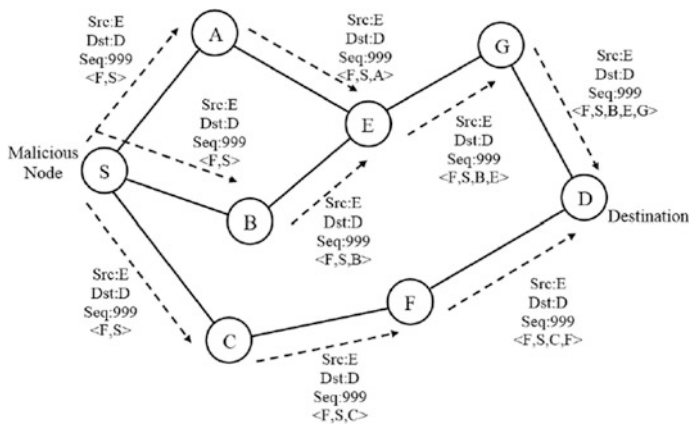
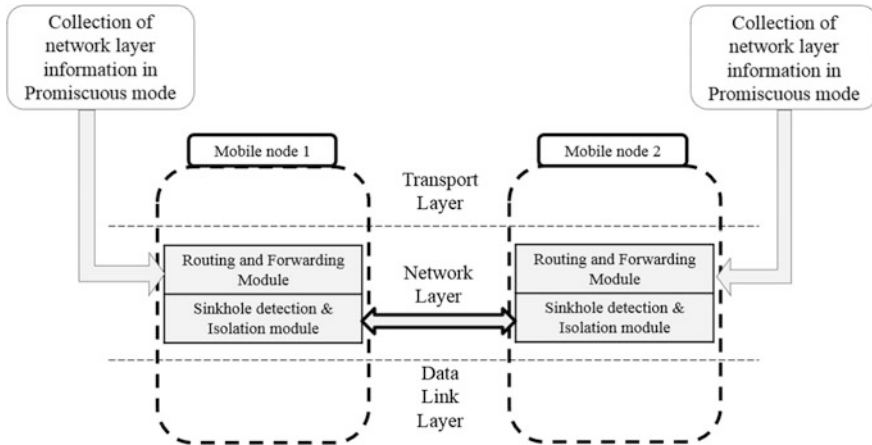


Fig. 1 Sinkhole attack by malicious node S



**Fig. 2** Collision-based sinkhole detection and isolation model

mechanisms. In network monitoring-based detection approaches [7–10], one or more mobile node is designated as monitor. The monitor notices the behavior of neighboring mobile nodes. Any suspicious sinkhole behavior is identified, and other nodes in the network are alerted. These methods exhibit heavy computational overhead on the monitoring node. Few other interesting works for the detection and removal of sinkhole attack demand modification in the routing protocols [11–15]. The underlying routing protocols are modified to capture the suspicious activities. This brings down the performance of the cloud. The agent-based detection methodologies proposed by [16, 17] require some centralized control for isolating sinkhole nodes. Hence, they are not suitable for the real-time implementation of the mobile ad hoc cloud network. Anomaly-based sinkhole detection and misuse detection methods are found in [18]. However, the topology of the MANET changes rapidly and the differences in the network state grow rapidly with time. Hence, anomaly-based detection schemes are not feasible for mobile ad hoc cloud. [14, 19, 20] have proposed intrusion detection schemes (IDS) for the detection of sinkhole attack. All the IDSs require central control in the cloud network. Further IDS schemes are vulnerable to single point of failures. Hence, IDS-based detection schemes are not desirable for mobile ad hoc cloud environment. Cryptographic functions are used extensively in approaches such as [21]. Cryptographic calculations demand higher processing power, and they are less desirable in the resource-constrained mobile cloud. Trust-based mechanisms recommended by [22, 23] are also not suitable for eliminating sinkhole attackers in mobile ad hoc cloud environment.

### 4 Proposed Methodology

The sinkhole node sends out fake RREQ messages in large. The frequency of RREQ message by a sinkhole node will be abnormally high. This is one of the suspicious parameters to detect sinkhole node. It also takes part in data transmission by sending route reply messages immediately. This route request and reply messages are monitored, and sinkhole node is isolated. The architecture of node collusion methodology for sinkhole detection and isolation is shown in Fig. 2.

Every node observes the routing information passed by its neighboring node. This routing information is fed into the detection and isolation module. This module is responsible for the detection of sinkhole nodes. Once the sinkhole node is detected, it is isolated by passing target alert message to the neighboring node. Let us consider that there are N nodes in the network denoted by N1, N2, ..., Ni. Every node Ni observes the RREQ message received over the time window Wt. Every node maintains a suspicious counter for other nodes. Whenever a node receives RREQ message, it increments counter corresponding to that node.

The sinkhole node has inherent characteristics of including itself in every routing information, and its identity is present in all the route request and reply messages. At time window W, every node checks its counter database. If it finds any node whose request has crossed the threshold λ, that node is delineated as sinkhole node. Any node that senses this abnormal behavior sends ‘target alert message’ (TAM) alerting other nodes about the possible sinkhole node and its victims.

The target alert message format is shown in Fig. 3.

The type field indicates the type of detection message used in node collusion technique.

#### Pseudocode 1: Sinkhole node detection

00	Let $N_i$ denote the mobile node
01	$SC_x^t$ denote the Suspicious Count of mobile node $x$ at the interval $t$ ;
02	$route_{rreq}^t$ denote the route specified in RREQ at the interval $t$ ;
03	for each window $\omega_t$ in the interval $t = 1, 2, 3, \dots, T$
04	for every $N_i$ during $\omega_t$
05	if ( $N_x \in route_{rreq}^t$ && $N_x = penultimate\ hop(route_{rreq}^t)$ ) then
06	$SC_x^t = SC_x^t + 1$
07	end if
08	end for
09	if ( $SC_x^t > \lambda$ ) then
10	$N_x$ is malicious; propagate TAM
11	end if
12	end for

The TAM message informs the network about the sequence number of the bogus route request and also the route affected by the sinkhole attack. Address[i] holds the

Type (8 bits)	Option Len (8 bits)	Identification (16 bits)
Sequence Number of Bogus RREQ		Sequence Number of TAM
Address [1]		
Address [2]		
....		
Address[n]		

**Fig. 3** Target alert message format

Type (8 bits)	Option Len (8 bits)	Identification (16 bits)
ID of Attacker Node		
ID of TAM		

**Fig. 4** Sinkhole information message format

address of nodes recorded in route. The routing information available in TAM is not updated in the routing cache. The sinkhole path is identified. For example, as shown in Fig. 2, suppose node C receives a TAM created by node F. If the routing information in C is  $\langle F, S, C \rangle$  and the route available in TAM is  $\langle F, S, C, F \rangle$ , then the common part is  $\langle F, S, C \rangle$ . This common path needs to be evaluated for the presence of sinkhole node.

Suppose that ‘n’ denotes the number of mobile nodes in the common path and ‘t’ denotes time interval. After (n, t) seconds, the node initiates sinkhole information message (SIM) in the network. The SIM format is presented in Fig. 4.

Every node that receives the SIM is prohibited from generating and transmitting another SIM in the network. The SIM contains information about the attackers’ ID and ID of the TAM.

## 5 Simulation Environment

The experiment was carried out using NS-2 simulator. DSR is used as routing protocol. The proposed method is compared with state of art literatures MDSR technique [24] and DSR-CM [12].

The performance of the node collusion method is evaluated by the following evaluation parameters:

- Packet delivery ratio (PDR): cumulative packets delivered successfully out of the transmitted packets.
- Routing overhead (RO): cumulative routing control messages traversed in the cloud.
- False-positive ratio (FPR): ratio of legitimate nodes that are wrongly stated as sinkhole nodes.

- Detection time (DT): time taken to isolate sinkhole node.

The simulation parameters are given in Table 1. The simulation was carried out for a period of 300 s.

## 6 Performance Analysis

### 6.1 Performance Analysis on Routing Overhead and Packet Delivery Ratio

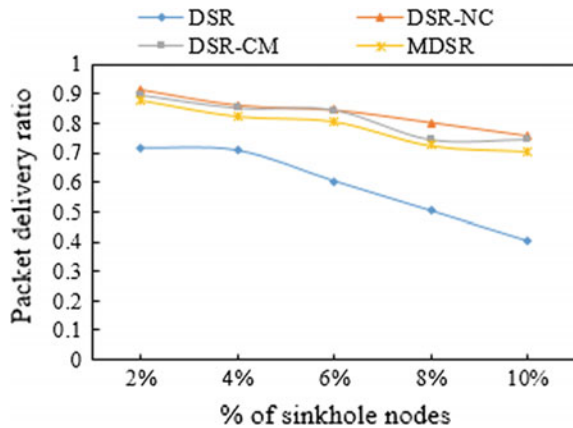
Initially, the routing overhead and packet delivery ratio were obtained by varying the percentage of sinkhole nodes. Two percentage of nodes are made as sinkhole nodes, and the performance parameters are measured. Then, the number of sinkhole node is varied from 4 to 10%. The results are shown in Figs. 5 and 6.

It is deduced from Fig. 6 that the PDR has improved for the node collusion method compared with MDSR [25] and DSR-CM [24] methods. The packet delivery ratio has improved by 2.32%. However, there is a decline in packet

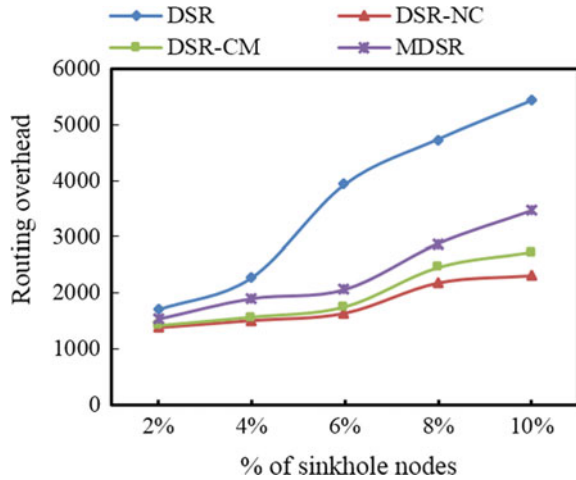
**Table 1** Simulation parameters

Parameter	Value
Topology area	750 × 750 m
Mobility parameter	Random waypoint model
Application used	5 kb UDP-CBR data payload: 512 bytes
No. of connections	20 pairs (40 nodes)
Type of traffic	UDP-CBR
Range of transmission	250 m

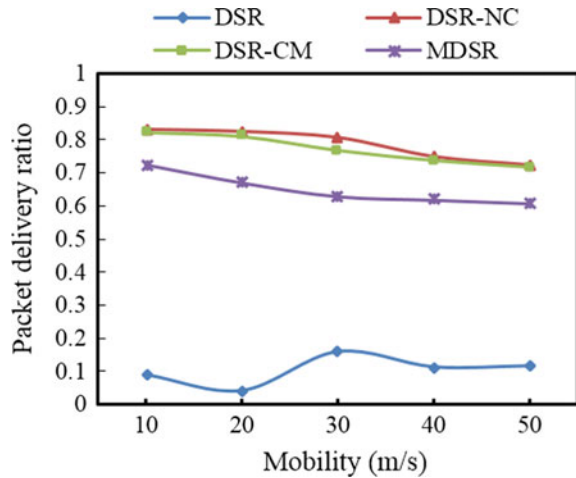
**Fig. 5** PDR versus percentage of sinkhole nodes



**Fig. 6** RO versus percentage of sinkhole nodes



**Fig. 7** PDR versus mobility in m/s

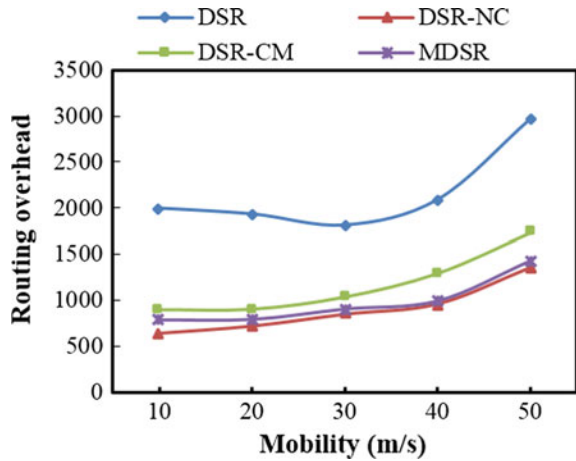


delivery ratio with the increase in the percentage of sinkhole nodes. Because the sinkhole nodes are eliminated from the cloud, it causes short downtime for the mobile users.

The mobile users have to search for a new route to reach destination eliminating the sinkhole node. In terms of control messages, node collusion method brings down the routing overhead when compared to MDSR and DSR-CM. The minimum routing overhead is due to the fact that the control messages are communicated only to the neighbors.

The packet delivery ratio and routing overhead are also measured by varying the mobility of the nodes. In this case, sinkhole attack is stimulated to be 10% of nodes; mobility is ranging from 10 to 50 m/s for a simulation period of 300 s. The results are shown in Figs. 7 and 8. The DSR-NC technique outperforms the existing

**Fig. 8** RO versus mobility in m/s



techniques, and the packet delivery ratio is improved by 2.40% and the routing overhead got reduced by 29.32%. This is because the node collusion technique detects and isolates the sinkhole nodes very fast which improves the critical performance metrics.

### 6.2 Performance Analysis on False-Positive Ratio and Detection Time

To investigate the false-positive ratio and detection time of the proposed methodology, a MANET with 200 nodes with the mobility of 20 m/s is simulated. The results are given in Table 2. It shows that the false-positive ratio of node collusion method is reduced by 18.67% when compared with DSR-CM. This is because the proposed method identifies sinkhole attackers and raises the sinkhole alarm only after the verification of the same with its neighborhood.

**Table 2** False-positive ratio and detection time

Percentage of sinkhole nodes (%)	False-positive ratio			Detection time		
	MDSR	DSR-CM	DSR-NC	MDSR	DSR-CM	DSR-NC
2	0.09865	0.0011	<b>0.00086</b>	3.9843	3.2129	<b>3.0453</b>
4	0.11401	0.05324	<b>0.04837</b>	5.3562	3.9299	<b>3.6336</b>
6	0.1224	0.07867	<b>0.06957</b>	5.5366	4.1324	<b>3.8911</b>
8	0.16758	0.12467	<b>0.10222</b>	6.1236	5.8219	<b>5.1256</b>
10	0.19443	0.17496	<b>0.14536</b>	6.8576	6.7226	<b>5.5629</b>

Similarly, the detection time in the proposed node collusion method is reduced by 10.8% when compared to DSR-CM. This improvement is because the nodes collude only with its neighboring node to conclude the sinkhole behavior of the attacker.

## 7 Conclusion

The node collusion technique has been proposed and simulated using DSR protocol. It has been observed that the proposed technique not only detects sinkhole node but also isolates the node from the network. The performance of the protocol is comparatively high with the existing techniques in the literature. The proposed technique also proved to be a lightweight technique as it does not include too much of control messages. The performance results such as packet delivery ratio, routing overhead, and the false-positive detection ratio also proved to be the best in the proposed technique.

This type of node collusion technique can be used to detect sinkhole node in tactical networks like war field, where sensitive information is always prone to passive and active attacks. This technique also preserves the security and privacy of the network by detecting and isolating the nodes. In future, extension work can be carried out to improve the false-positive ratio.

## Reference

1. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. *Future Gener Comput Syst* **29**(1), 84–106 (2013)
2. Huerta-Canepa, G., Lee, D.: A virtual cloud computing provider for mobile devices. In: *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond*. ACM (2010)
3. Rahimi, M.Reza, et al.: Mobile cloud computing: a survey, state of art and future directions. *Mobile Netw. Appl.* **19**(2), 133–143 (2014)
4. Dinh, H.T., et al.: A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel. Commun. Mobile Comput.* **13**(18), 1587–1611 (2013)
5. Othman, M., Xia, F., Abdul Nasir Khan: Context-aware mobile cloud computing and its challenges. *IEEE Cloud Comput.* **2**(3), 42–49 (2015)
6. Yaqoob, I., et al.: Mobile ad hoc cloud: a survey. *Wirel. Commun. Mobile Comput.* **16**(16), 2572–2589 (2016)
7. Bansal, S., Baker, M.: Observation-based cooperation enforcement in ad hoc networks. Research Report cs. NI/ 0307012, Stanford University
8. Gandhewar, N., Patel, R.: Detection and prevention of sinkhole attack on AODV protocol in mobile adhoc network. In: *2012 Fourth International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 714–718. IEEE (2012)
9. Gagandeep, A., Kumar, P.: Analysis of different security attacks in MANETs on protocol stack A-review. *Int. J. Eng. Adv. Technol. (IJEAT)* **1**(5), 269–275 (2012)



10. Kim, G., Han, Y., Kim, S.: A cooperative-sinkhole detection method for mobile ad hoc networks. *Int. J. Electron. Commun.* **64**, 390–397 (2010)
11. Girish Kumar, V., Rachit Jain: A table driven search approach for revelation and anticipation of sinkhole attack in MANET. *Int. J. Res. Eng. Technol.* **05**(08), 20–25 (2016)
12. Li, X., Jia, Z., Zhang, P., Zhang, R., Wang, H.: Trust-based on-demand multipath routing in mobile ad hoc networks. *IET Inf. Secur.* **4**(4), 212–232 (2010)
13. Mitchell, R., Chen, R.: A survey of intrusion detection in wireless network applications. *Comput. Commun.* **42**, 1–23 (2014)
14. Sen, S., Clark, J.A., Tapiador, J.E.: Power-aware intrusion detection in mobile ad hoc networks. In: *Proceedings of the Ad Hoc Networks. Lecture Notes of the Institute of Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 28, pp. 224–239. Springer (2010)
15. Vishnu, K., Paul, A.J.: Detection and removal of cooperative black/gray hole attack in mobile ad hoc networks. *Int. J. Comput. Appl.* **1**(22), 38–42 (2010)
16. Stafrace, S.K., Antonopoulos, N.: Military tactics in agent-based sinkhole attack detection for wireless ad hoc networks. *Comput. Commun.* **33**, 619–638 (2010)
17. Shafiei, H., Khonsari, A., Derakhshi, H., Mousavi, P.: Detection and mitigation of sinkhole attacks in wireless sensor networks. *J. Comput. Syst. Sci.* **80**(3), 644–653 (2014)
18. Sarika, S., Pravin, A., Vijayakumar, A., Selvamani, K.: Security issues in mobile ad hoc networks. *Proc. Comput. Sci.* **92**, 329–335 (2016)
19. Su, M.-Y.: Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. *Comput. Commun.* **34**, 107–117 (2011)
20. Mitrokotsa, A., Dimitrakakis, C.: Intrusion detection in MANET using classification algorithms: the effects of cost and model selection. *Ad Hoc Netw.* **11**(1), 226–237 (2013)
21. Vennila, G., Arivazhagan, D., Manickasankari, N.: Prevention of co-operative black hole attack in manet on DSR protocol using cryptographic algorithm. *Int. J. Eng. Technol. (IJET)* **6**(5), 2401 (2014)
22. Sanchez-Casado, L., Macia-Fernandez, G., Garcia-Teodoro, P., Aschenbruck, N.: Identification of contamination zones for sinkhole detection in MANETs. *J. Netw. Comput. Appl.* **54**, 62–77 (2015)
23. Thanachai, T., Tapanan, Y., Punthep, S.: Adaptive sinkhole detection on wireless ad hoc networks. In: *Proceedings of IEEE aerospace conference*, 4–11. IEEE, NJ, USA (2006)
24. Kim, G., Han, Y., Kim, S.: A cooperative-sinkhole detection method for mobile ad hoc networks. *AEU-Int. J. Electron. Commun.* **64**(5), 390–397 (2010)
25. Mohanapriya, M., Krishnamurthi, I.: Modified DSR protocol for detection and removal of selective black hole attack in MANET. *Comput. Electr. Eng.* **40**(2), 530–538 (2014)

# Asymmetric Addition Chaining Cryptographic Algorithm (ACCA) for Data Security in Cloud



D. I. George Amalarethinam and H. M. Leena

**Abstract** Cloud computing is a centralized network which enables the movement of data and application software and access the resources on demand. Although this type of computing meets the need of the users, various issues present in the cloud decreases the usage of resources. Among them, security is one of the major issues. The users move to the cloud for storing their sensitive data. Thus, data protection is a key concern. Cryptography is the conventional method applied for securing data in an efficient way. Encryption and decryption processes of cryptography proposed by various predefined algorithms take much time. Thus, the concentration must be given to reduce the time of these processes. Many methodologies exist for reducing the time which is more mathematical in nature. One among them is the concept of addition chaining method. The proposed asymmetric cryptographic algorithm uses the concept of addition chaining to reduce the time spent both in encryption and decryption processes.

**Keywords** Cloud computing • Data security • Cryptography  
RSA • Addition chaining

## 1 Introduction

Cloud computing is a distributed collection of interconnected and related systems with more resources provisioned on the basis of pay-per-use. It is more related to other computing technologies like grid, utility, and autonomic computing. The Cloud is different from others especially the grid, by its virtualization technology. This technology leverage better resource utilization and dynamic resource provisioning.

---

D. I. George Amalarethinam  
Jamal Mohamed College (Autonomous), Trichy 620020, Tamilnadu, India  
e-mail: di\_george@gmail.com

H. M. Leena (✉)  
Holy Cross College (Autonomous), Trichy 620002, Tamilnadu, India  
e-mail: leena\_raja@yahoo.co.in

Both the users as well as the providers are benefitted from cloud computing: providers can reuse the resources while the users can acquire and deploy the resources based on their needs. Since it is based on pay-per-use model, nowadays a wide range of users or customers rely more on cloud for the betterment of resource usage in a cost effective manner. Even though IT enterprises are engaged with more cloud technologies, there are some issues which make the customers to move to reduced usage of the resources. The issues start from security, availability, performance, lack of standards, increased usage cost, and struggle in integrating with on-premise IT resources. So many customers do not fully depend on the cloud environment because of their unawareness of the location where their data would be stored, its availability in their need, and data processing by others. For these reasons many of the researchers concentrates on the “security” aspect of the cloud. Security has its own underlings like data security and privacy, disaster recovery, management of identity and access, business continuity planning, compliance issues, challenges in migration, and more.

Most of the customers’ data and business logic are preferably stored in the cloud servers which are remotely accessed. Users’ data include some sensitive information like financial, health records, and even personal information which may be disclosed to public. So data security is the major challenge among the prescribed issues under security [1]. The conventional method to secure the data of the users is cryptography. Various algorithms for encrypting and decrypting the data are categorized under symmetric and asymmetric. Symmetric algorithms use the same key both for encryption and decryption processes where asymmetric concentrates on two different keys for these processes. The major concern regarding these processes is the time consumption. The users cannot wait much time for these processes, and it takes extra time to write the data into cloud storage after encryption. Thus, many researchers turn their focus to reduce the time spent for encryption and decryption.

## 2 Related Work

Li and Ma [2] created a database of addition chains. This improved the efficiency of the modular exponentiation calculation of RSA algorithm. Reducing the times of multiplication enhance the speed of modular exponentiation calculation. This can be achieved by using any of the algorithms like binary algorithm, slide window algorithm, coding algorithm, addition chaining algorithm. Thus, the author focused on the addition chaining algorithm for reducing time of execution of RSA algorithm.

Mani [3] proposed division-based addition chain which generates the optimal addition chain for the small exponents. For large exponents, there is a small increase in chains length. This reduces the encryption and decryption time for most of the asymmetric key algorithms. The proposed work is also very simple, deterministic, and based on division.

Domínguez-Isidro and Mezura-Montes [4] proposed an evolutionary programming algorithm to find the minimum addition chain. The core of the algorithm

consists of suitable fitness function and initial population, a mutation operator, and the survivor selection mechanism. The algorithm was able to generate better accumulated addition chains with respect to three state-of-the-art algorithms.

Clift [5] calculated optimal addition chains and a proposed new algorithm. The algorithm is faster than the previous known methods. The lengths of all optimal addition chains for  $n \leq 32$  were calculated, and the conjecture was disproved.

Mezura-Montes et al. [6] proposed an algorithm to solve the minimal length addition chain problem. This algorithm is based only on a mutation operator and a stochastic replacement process to bias the search to competitive solutions by requiring less evaluation per run.

### 3 Proposed Work

The proposed ACCA algorithm uses the mathematical concept called addition chaining to reduce the time spent for encryption and decryption. The enhanced RSA (ERSA) [7] algorithm uses two extra prime numbers along with the existing ones. Two “N” values namely “N1” and “N2” are calculated with the multi-prime numbers. The ERSA multi-prime algorithm uses these “N” values for finding out the public and private keys “E” and “D.” This algorithm gets the input in terms of file. The file sizes extended up to 64 KB. But it uses a fixed size of key. The algorithm is further revised by splitting the file into blocks with varying sizes of key like 128, 256, 512 [8]. The size of the block is determined by the key size. Eq. 1 is proposed by George Amalarethinam et al. [9] and reflected as suitable block size.

$$\text{Block Size} = (2 * \text{Key Size}) - 1 \tag{1}$$

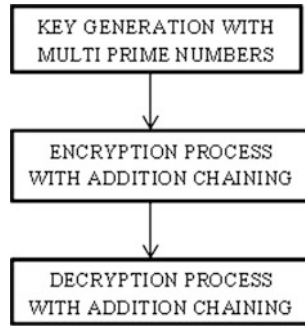
Therefore, Eq. 1 is considered here for calculating the block size. The results showed a good variation in reduction of time for both the encryption and decryption processes. Further enhancement is made in the proposed algorithm addition chaining cryptographic algorithm (ACCA) by using the mathematical concept called addition chaining. Addition chaining aims to minimize the number of modular multiplications and to optimize the individual modular multiplications [9]. The ACCA concentrates on minimizing the individual modular multiplication.

For example, the calculation of  $a^8 \bmod n$  can be done as  $(a * a * a * a * a * a * a * a) \bmod n$ .

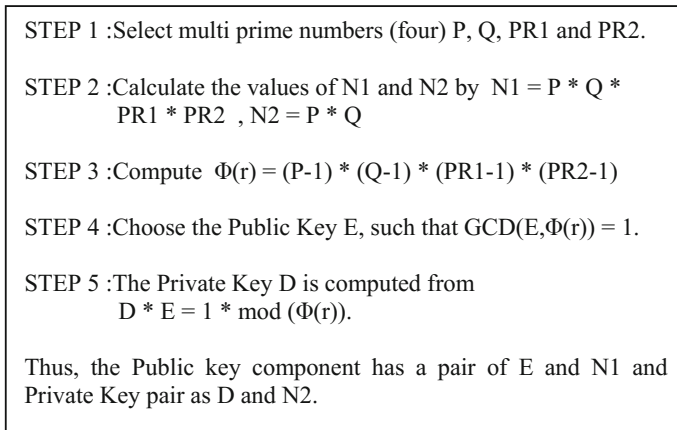
As an alternative, three smaller multiplications and modular reductions, i.e.,  $((a^2 \bmod n)^2 \bmod n)^2 \bmod n$  is used to bring out the same result. This is called addition chaining or the binary square and multiply method [9].

The same addition chaining method is used in the proposed algorithm ACCA for enhancing the encryption and decryption processes. The method is applied specifically for converting the plain text into ciphertext and vice versa.

The steps of the proposed algorithm ACCA is shown in Fig. 1.



**Fig. 1** Steps of the ACCA algorithm



**Fig. 2** Key generation with multi-prime numbers

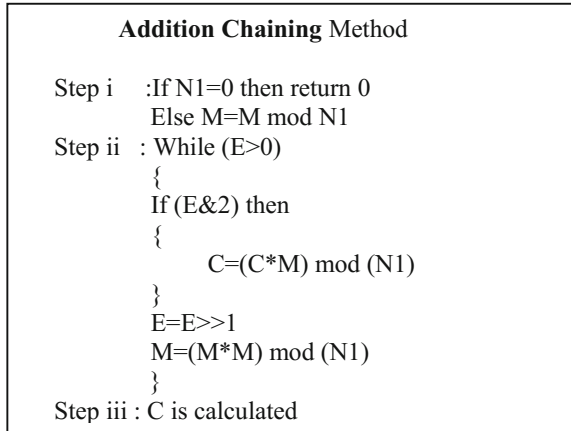
The ACCA algorithm is composed of three major steps. Each of these steps is explained in the following figures.

Figure 2 shows the process of key generation.

**Step 1** is the process of key generation. Generally, RSA algorithm uses two prime numbers. In addition, two more prime numbers, namely PR1 and PR2 are included in the proposed algorithm ERSA. The next step of the algorithm computes two “N” values such as N1 and N2. Four prime numbers are multiplied and computed as N1. For N2 computation, it uses two prime numbers. This multi-prime concept is used to increasing the complexity of the encryption part.

Figure 3 gives the steps of encryption process with addition chaining.

**Step 2 of ACCA algorithm** includes the encryption process. In RSA algorithm, the formula used to encrypt the plain text is  $C = M^E \text{ mod} (N1)$ . But this is replaced by addition chaining method in ACCA algorithm.

**Fig. 3** Encryption process with addition chaining

Step i of encryption process check whether calculated  $N1$  value is zero. If  $N1 = 0$ , then it will return 0. Otherwise, the modulus operation is performed.

Step ii starts by checking whether the exponent is greater than zero. The comparison is done on bigger number. The same exponent is also checked if modulus operation gives zero. This is replaced by the AND operator. If it is so, ciphertext calculation is done. If not, the exponent is shifted to right by 1. The shift right operation is done rather than the division operation, to reduce the time consumed by division operator. The plain text is multiplied by itself, and the resultant is applied to the modulus operator by  $N1$ . Step 2 is repeated until the exponent reaches zero.

Step iii gives the calculated value of ciphertext.

Here, the modular exponentiation operation is replaced by the mathematical calculation called addition chaining.

Specific operations like modulus and divide are replaced by “&” and shift operations to reduce the time.

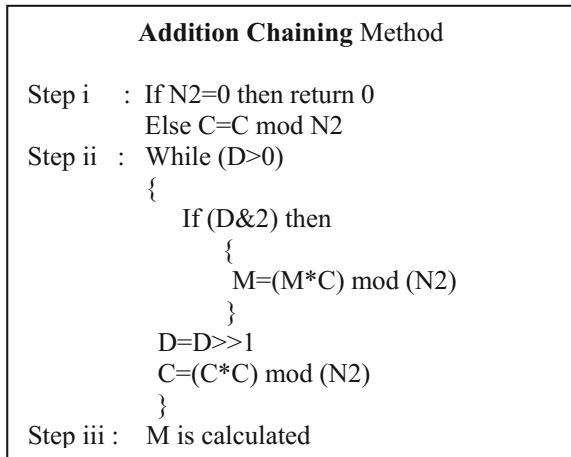
Figure 4 shows the steps of decryption process with addition chaining.

**Step 3** shows the calculation of decrypted message which is depicted in Fig. 4.

The steps of encryption process are repeated for decryption process with minor changes.

- i. In the place of  $M$ ,  $C$  is used and vice versa.
- ii. Instead of  $N1$ ,  $N2$  value is used.
- iii. Encryption key  $E$  is replaced by decryption key  $D$ .

**Fig. 4** Decryption process with addition chaining



This addition chaining concept is applied to encryption and decryption processes. It is tested with text file as input and the time is recorded. The same file which is divided into different blocks is given as the input and the time for encryption and decryption are recorded.

## 4 Illustration

The proposed algorithm ACCA-F test the file and ACCA-B test the file divided into blocks. An illustration is given below for the sample input string “**Hello welcome to the java world.**”

Key size chosen for this small string is 8.

$$\text{Block Size} = 2 * \text{KS} - 1 = 15$$

The string is given as input to ACCA-B, which divides them into blocks. The given input string is divided into blocks and each block has 15 characters based on the key size.

Thus, the block 1 contains the partitioned string: “Hello welcome t”.

### Step 1: Key generation with multi-prime numbers.

Step 1: Four prime numbers are generated

$$P = 197, Q = 223, PR1 = 191, PR2 = 167$$

Step 2:  $N1 = P*Q*PR1*PR2$ ,  $N2 = P*Q$

$$N1 = 1401267107, N2 = 43931$$

Step 3:  $\Phi(r) = (P - 1) * (Q - 1) * (PR1 - 1) * (PR2 - 1)$

$\Phi(r) = 1372368480$

Step 4: Public key (E): 101

Step 5: Private key (D): 203817101

## Step 2: Encryption process with addition chaining

```

Step i : If N1=0 then return 0
        Since N1= 1401267107 goes to else
        Else M=M mod N1
        M=1905742
Step ii : While (E>0)
        101 > 0
        {
            If (E&1) then
            {
                C=(C*M) mod (N1)
                C=150d5d6
            }
            E=E>>1; 101>>1=50
            M=(M*M) mod (N1)
            M=1169496327
        }

```

The above step is repeated until all the characters in a block are encrypted.

Step iii: C is calculated.

Sample encrypted message for the above block is

682345027 1253596686 469653816 469653816 1184459685 662431346 1092795030 1253596686  
469653816 783842438 1184459685 485970386 1253596686 662431346 565449362

Remaining 2 blocks of characters are encrypted with different prime numbers and key E.

**Encryption time for these 3 blocks is: 31 + 16 + 4 = 51 ms.**

## Step 3: Decryption process with addition chaining

The 3 steps of decryption process are executed in reverse order with the values of N2 and D, instead of N1 and E.

Decrypted Message of the Block 1 is "Hello welcome t". Remaining Encrypted messages of 2 Blocks are decrypted and shown as "o the java worl" and "d".

**Decryption time for these 3 blocks is: 25 + 10 + 5 = 40 ms.**

The same string is applied to ACCA algorithm as a file.

**Encryption time: 93 ms**

**Decryption time: 56 ms**



**Table 1** Comparison of encryption and decryption time of algorithms ACCA-F and ACCA-B with RSA

File size (KB)	Key size (bits)	Block size (bits)	Encryption time (ms)			Decryption time (ms)		
			RSA	ACCA-F	ACCA-B	RSA	ACCA-F	ACCA-B
32	64	127	40576	47408	11641	6567	21528	19780
	128	255	107906	155298	74462	35646	97781	92826
	256	511	646741	708335	535313	230116	616186	615998
	512	1023	4512721	5228555	4156330	1350580	4558313	4502006

### 5 Results and Discussions

The algorithm is compiled and run on Java environment version 7. Two algorithms are compared. The algorithm ACCA-F which uses file with addition chaining is slower than for encryption as well as decryption processes than ACCA-B that uses blocks with addition chaining. Both the algorithms are compared with the standard RSA algorithm.

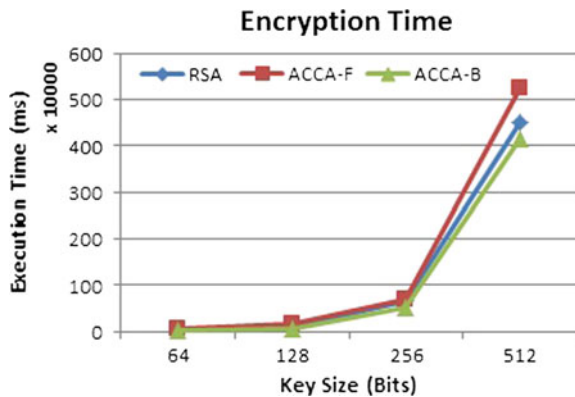
Table 1 shows this variation in time for both encryption and decryption.

It is observed that proposed algorithm ACCA-B outperforms the ACCA-F in encryption speed and shows better result in decryption time. ACCA-B also shows enhanced encryption time reduction than RSA algorithm.

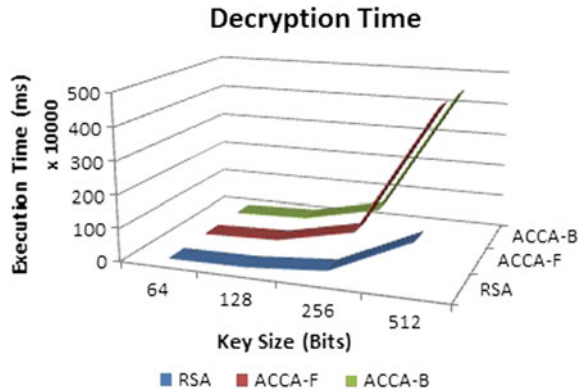
Figures 5 and 6 represent the reduction in encryption according to ACCA-B than ACCA-F and RSA. However, in decryption time there is a reduction according to ACCA-B than the ACCA-F algorithm.

Figs. 5 and 6 reveal that the encryption and decryption of the ACCA-B algorithm is faster than the ACCA-F algorithm. It shows that the usage of the concept of addition chaining has a great impact on the algorithm which divides the file into blocks than the algorithm which uses the whole file.

**Fig. 5** Comparison of encryption time between RSA, ACCA-F AND ACCA-B algorithms



**Fig. 6** Comparison of decryption time of RSA, ACCA-F AND ACCA-B algorithms



**Table 2** Comparison of average encryption and decryption time of RSA, ACCA-F, and ACCA-B algorithms

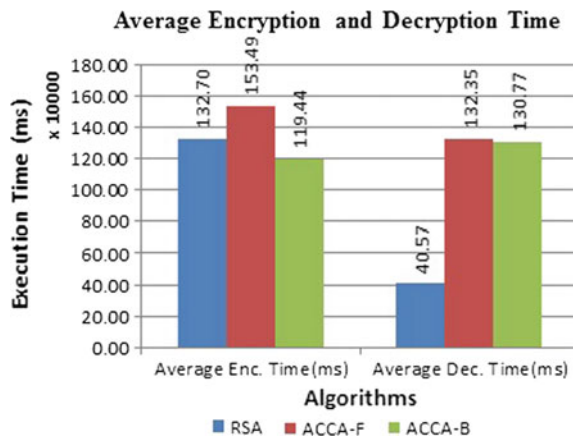
Algorithm	Average enc. time (ms)	Average dec. time (ms)
RSA	1,326,986.00	405,727.25
ACCA-F	1,534,899.00	1,323,452.00
ACCA-B	1,194,436.50	1,307,652.50

Figures 5 and 6 also imply that encryption process of ACCA-B is better than the other two algorithms, ACCA-F and the standard algorithm RSA. According to decryption time, the RSA algorithm shows better result than the ACCA-F and ACCA-B.

Figure 6 shows the improvement in decryption time with respect to ACCA-B algorithm than the ACCA-F algorithm.

Table 2 exposes the average encryption and decryption time for RSA, ACCA-F, and ACCA-B algorithms.

**Fig. 7** Comparison of average encryption and decryption time of ACCA-F and ACCA-B algorithms



From Table 2, it is revealed that the average encryption time of ACCA-B is much condensed than RSA and ACCA-F algorithms.

Figure 7 depicts the average encryption and decryption time of RSA, ACCA-F, and ACCA-B algorithms.

From Fig. 7, it is clearly revealed that the average encryption time of ACCA-B shows much improved result than ACCA-F. This performance of ACCA-B confirms that the concept of addition chaining used for the splitted blocks reflects a better change in average encryption and improved decryption time.

## 6 Conclusion and Future Work

The usage of addition chaining for encrypting and decrypting the messages reduce the time of both the processes. Further, when this concept used with blocks shows an enhanced performance in encryption time and improved result in decryption time. This paved the way for the cloud user to store their encrypted data in cloud storage quickly. The ACCA-F algorithm with addition chaining is slower than ACCA-B for both encryption and decryption processes. The results imply that the security level of encryption process is enhanced. This has been tested with smaller files. In future, the enhancement of the security level is to be analyzed while reducing the overall decryption time.

## References

1. Aggarwal, N., Tyagi, P., Dubey, B.P., Pilli, E.S.: Cloud computing: data storage security analysis and its challenges. *Int. J. Comput. Appl.* **70**, 33–37 (2013)
2. Li, Y., Ma, Q.: Design and implementation of layer extended shortest addition chains database for fast modular exponentiation in RSA. In: *International Conference on Web Information Systems and Mining*, pp. 136–139 (2010)
3. Mani, K.: Generation of addition chain using deterministic division based method. *Int. J. Comput. Sci. Eng. Technol.* **4**, 553–560 (2013)
4. Domínguez-Isidro, S., Mezura-Montes, E.: An evolutionary programming algorithm to find minimal addition chains. **22** (2011)
5. Clift, N.M.: Calculating optimal addition chains. Open access—Springerlink.com (2010)
6. Mezura-Montes, E., Domínguez-Isidro, S., Osorio-Hernández, L.G.: Addition chain length minimization with evolutionary programming. *ACM* (2011)
7. George Amalarethinam, D.I., Leena, H.M.: Enhanced RSA algorithm for data security in cloud. *Int. J. Control Theor. Appl.* **9**, 147–152 (2016)
8. George Amalarethinam, D.I., Leena, H.M.: Enhanced RSA algorithm with varying key sizes for data security in cloud. *IEEE Xplore Library* (2017)
9. George Amalarethinam, D.I., Sai Geetha, J., Mani, K.: Analysis and enhancement of speed in public key cryptography using message encoding algorithm. *Indian J. Sci. Technol.* **8**, 1–7 (2015)

# Improved Key Generation Scheme of RSA (IKGSR) Algorithm Based on Offline Storage for Cloud



P. Chinnasamy and P. Deepalakshmi

**Abstract** Cloud computing is the most prominently used technology to store and recover information from anyplace with the assistance of web. One of the challenging tasks is to provide security to client information stored in cloud. To enhance the security of clients information, in this paper, (IKGSR) an improved key generation scheme for RSA algorithm is introduced which employs four giant prime numbers to create encoding (E) and decoding (D) keys. A database schema is also designed to save the key elements of RSA before it initiates.

**Keywords** Cloud security · Indexes · Offline storage · RSA algorithm  
Key generation · Cloud storage

## 1 Introduction

Cloud computing is one of the most dominant and growing technologies in Information Technology and useful in many day-to-day applications like booking train tickets and getting birth and death certificates with the help of internet [1, 2]. In case of storing personal or official information, security is the primary and a major factor to adopt cloud computing in real-time applications. There are a number of security threats to cloud such as trustworthy access control, identity management, insecure application programming interface, insider threats, VM hyperjacking, improper encryption and decryption, and VM Images [3].

---

P. Chinnasamy (✉) · P. Deepalakshmi  
Kalasalingam University, Krishnankoil, Srivilliputtur 626126, Tamilnadu, India  
e-mail: chinnasamyponnusamy@gmail.com

P. Deepalakshmi  
e-mail: deepa.kumar@klu.ac.in

Cryptography [4] is a common method to provide security and ensure CIA concepts (Confidentiality, Integrity, and Authenticity). A message, a stream of bits, a bitmap, a stream of digitized voice, a digital video image, etc., readable by an attacker is known as plaintext, M. The message altered to be unreadable by anyone except the intended recipients with an appropriate cryptography algorithm is known as ciphertext, C. Based on the style in which encryption and decryption is carried, there are two types of cryptosystems, namely symmetric cryptography and asymmetric cryptography. In symmetric algorithms, like DES, 3DES, IDEA, and BLOWFISH, encipher and decipher are performed using the distinct keys. In asymmetric algorithms, like RSA and elliptic-curve cryptography (ECC), encryption and decryption are performed by two different keys. Among the three phases of RSA [4, 5], in first phase, key generation is finished, and in second phase, the initial plaintext is reborn into ciphertext in sender side and in third phase, ciphertext is born again into plaintext in receiver side. In this proposed work, an interchangeable database over the network is used to hold the key elements of RSA algorithm into two totally different database tables before starting any crypt process. Also, associate index value is used instead of primary values for public- (e) and personal-keys (d).

The rest of this paper is structured as follows. In Sect. 2, RSA variants are discussed along with merits and demerits. Section 3 presents proposed algorithm in detail. In Sect. 4, results of proposed technique are dealt in detail, and conclusion is given in Sect. 5.

## 2 Related Works

Somani and Mangal [6] utilized three prime numbers to enhance the security level at decryption side to stay away from some security assaults on RSA. They introduced a complex analysis to find the modulus factor  $n$ . The main feature here was the usage of CRT (Chinese Remainder Theorem) to reduce the decryption time.

Patidar and Bhartiya [7] modified RSA cryptosystem to support offline storage and prime quantity. This improved the information interchange across unsecured networks by making use of three giant prime numbers, which made the public element difficult to factorize by an interrupter. A database system was also employed to hold key elements of RSA.

Jamegar and Joshi [5] proposed secure RSA, which can be used for secure file transmission. Abd et al. [8] surveyed about the essential security advantages consisting of traditional security challenges over a cloud computing model by both providers and users of cloud.

Song and Chen [9] found a unique approach to well-developed form of hidden mapping hyper combined public-key management scheme supported by hyper elliptic curve (HEC) crypto system. It solved the matter of enormous scale key management and storage problems in cloud with high proficiency and capability. It resisted collusion attack, and guaranteed safe and reliable service to be provided by the cloud storage services.

Buchade and Ingle [2] stated a new method called key management for cloud data storage. They compared different key management methods to different cloud environments and evaluated symmetric key algorithms and concluded the limitations as key unavailability, key destruction, key distribution, and key updating.

Thangavel et al. [10] proposed increased and secured RSA key generation scheme (ESRKGS) by utilizing four substantial prime numbers to create public- and personal-keys. However, secret writing and decryption were performed by  $n$ , a multiply of two prime numbers. They tried to interrupt their frameworks through brute-force attack and results showed that time to find  $n$  was very high compared to that of original RSA.

Luy et al. [11] proposed an enhanced and secured RSA key generation scheme (ESRKGS) method in which they chose alternate private key to break the security system but results indicated that ESKGSR security level was almost as same as in the traditional RSA algorithm.

In general, RSA algorithm works step by step compared to different asymmetric algorithms and provides not as much of protection over the communications channel. To boost the speed of RSA computation and to enhance the safety level, in this paper, the RSA algorithmic rule is reworked by four giant prime numbers and calculated the public element  $n$ , which needs to be the result of three prime numbers.

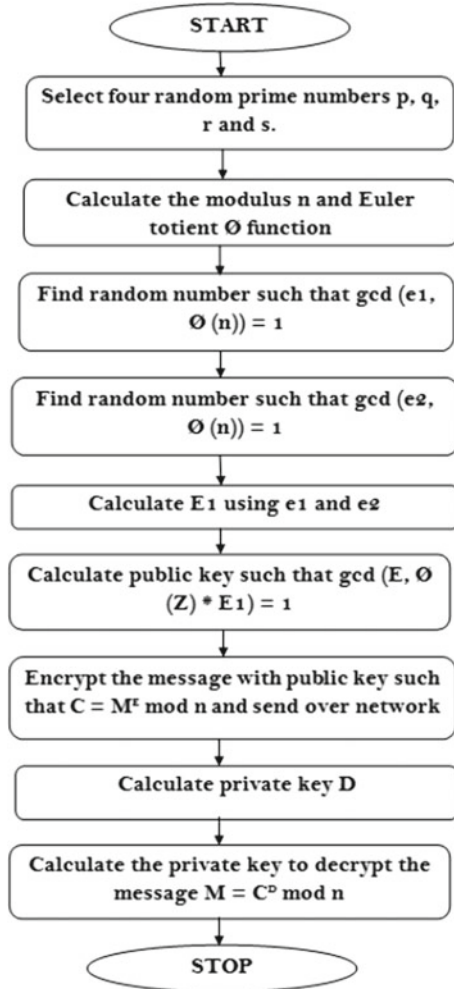
### 3 Proposed Framework

The proposed model shown in Fig. 1 is designed depending on an enhanced key setup of RSA cryptosystem [12]. For proposed algorithm,  $p$ ,  $q$ ,  $r$ , and  $s$  are assumed as prime numbers,  $n$  is common modulus,  $E$  is public-key,  $D$  is secret key, and  $M$  is the original message. The factor of  $E$ ,  $D$  depends upon  $Z$ , which is the product of four prime numbers considered. The value of  $E$  is not directly calculated, but before finding  $E$ ,  $E1$  has to be calculated which relies on two totally different values,  $e1$  and  $e2$ .

#### 3.1 Offline Storage of Proposed Framework

The security and speed of RSA algorithm can be improved through offline storage of key elements as shown in Fig. 2. The key sets of RSA are saved in database schema that is exclusive to all or any network. All related parameters are stored in database

**Fig. 1** Flowchart of proposed technique



before initiating algorithmic process. There are two tables within a database to keep the key parameters. First table will have the values of  $p, q, N1, (N)$ , and  $n$ ; and second one will have the values of  $e, d, r, s, E1$ , and  $D1$ . Anybody trying to guess the worth of modulus  $n$  can not get action since the worth of  $n$  depends on results of the three prime numbers as  $n = p * q * r$ . Also, it is troublesome to break each table at the same time.  $E1$  and  $D1$  are the indexes of public- and personal-key.

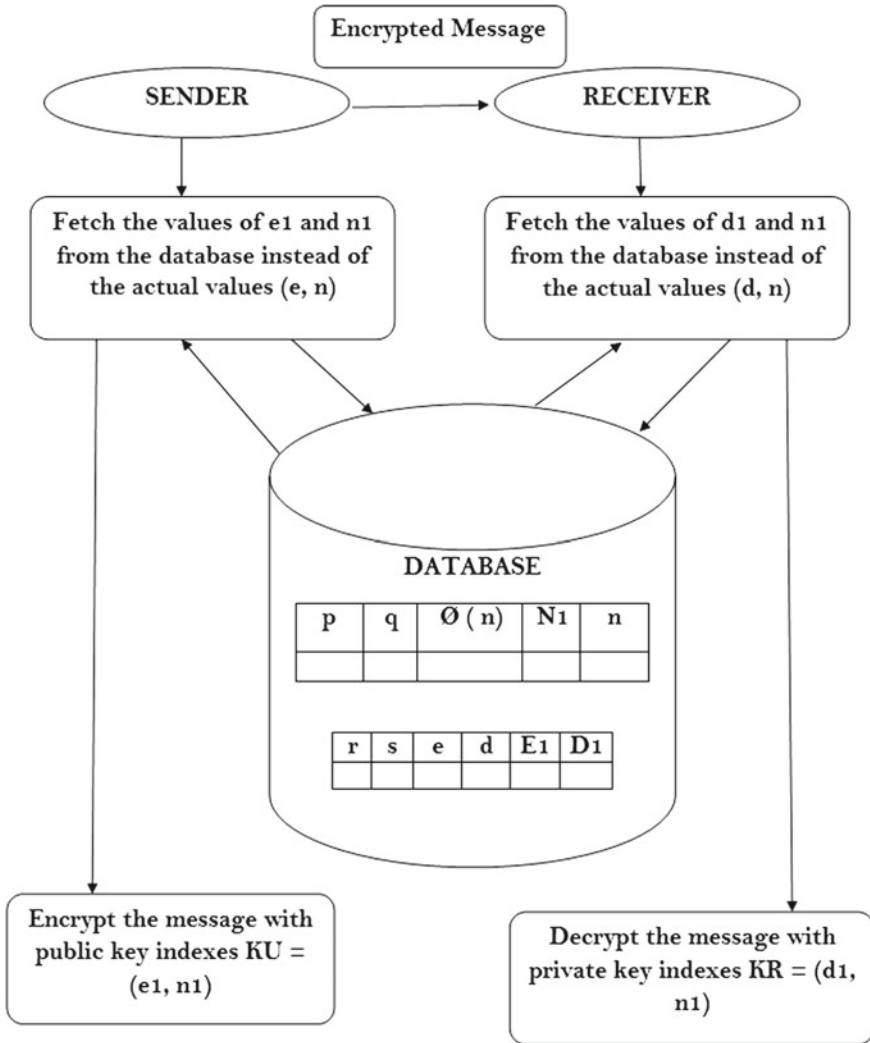


Fig. 2 Architecture of proposed technique

### 4 Implementation Details

Proposed work was implemented using Java Big Integer Library since it has special functions like finding GCD of numbers, prime number checking, and prime number generation. Here, client has to either enter prime numbers or specify the bit length of prime numbers to generate random prime numbers. A SQLite database engine is utilized to store the key parameters of proposed framework, which was implemented



using JAVA (Jdk 1.7), SQLite database, which are running on a 3.10 GHz, Intel Core i5-4440 C.P.U @ 3.10 GHz processor, 8 GB RAM, and Ubuntu 14.04 (64 Bit) Operating System.

#### 4.1 Performance Analysis

The performance of basic RSA as RSA1 and ESRKGSA as RSA2 is shown in Tables 1 and 2. The proposed methodology was tested for varying length of inputs. The performance of the proposed method, IKGSR, was measured in terms of key formation time, encipher and decipher time, and results are shown in Table 3. The time for key formation of the proposed methodology is somewhat higher than that of both RSA1 and RSA2. This enhanced key formation time makes the system demanding hard to interrupt. The timer required for encryption and decryption is shown in Table 3 for increased prime number size in terms of bits.

**Table 1** Performance of RSA1

Range of primes (in bits)	Key formation time (in ms)	Encipher time (in ms)	Decipher time (in ms)	Total time (in ms)
100	40	1	1	42
128	42	1.1	2	45
256	95	14	7	116
512	171	35	40	246
1024	605	136	249	990
2048	6861	922	1818	9601
4096	41801	6740	13203	61744

**Table 2** Performance of RSA2

Range of primes (in bits)	Key formation time (in ms)	Encipher time (in ms)	Decipher time (in ms)	Total time (in ms)
100	47	1	2	50
128	58	1.2	2	70
256	97	11	15	123
512	234	22	73	329
1024	936	125	515	1576
2048	7981	900	3552	12433
4096	80872	6808	26704	114384

**Table 3** Performance of improved key generation scheme of RSA (IKGSR)

Range of primes (in bits)	Key formation time (in ms)	Encipher time (in ms)	Decipher time (in ms)	Total time (in ms)
100	60	1	2	63
128	65	2	2	69
256	131	13	12	156
512	272	25	79	376
1024	1412	140	530	2082
2048	14511	977	3933	19421
4096	166020	7080	28043	201143

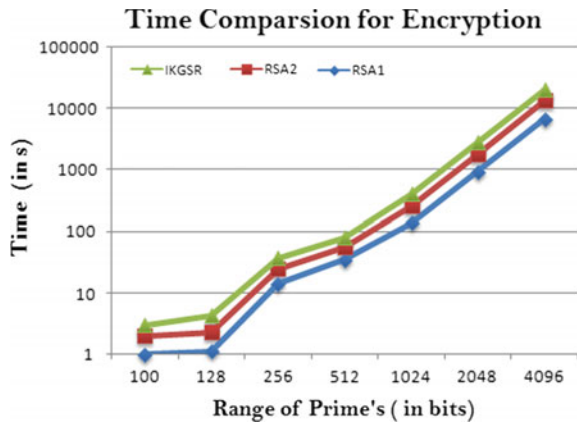
### 4.2 Security Analysis

Four different attacks normally used to interrupt RSA algorithms are as follows: brute-force, mathematical attacks, timing attacks, and chosen ciphertext attacks. Brute-force method is used for security analysis of the work proposed here.

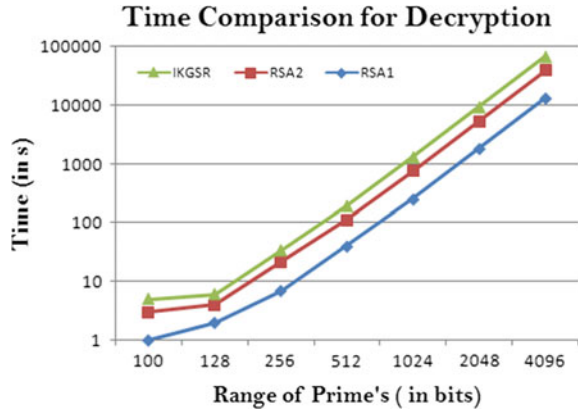
The time to interrupt RSA system is akin to the time taken to find the prime components used. This means the time needed for factoring of common modulus  $n$ . In general, elliptic-curve factorization (ECM) and general number field sieve (GNFS) are the preferred factorization methods. For smaller range factorization, we can select ECM, whereas for integers bigger than hundred digits, we can select GNFS factoring methodology. Fermat factorization is used to break secret code-supported numbers. But in the proposed methodology,  $n$  is based on product of the 3 prime numbers, and this makes search of personal-key,  $D$  tough. These two techniques will be used to find  $p$  and  $q$ ; however, other two primes can be found only by using brute-force.

The encryption time and decryption time comparison of basic RSAs and proposed method are clearly shown in Fig. 3 and Fig. 4, respectively.

**Fig. 3** Time comparison for encryption



**Fig. 4** Time comparison for decryption



Specifically,

$$T_{setup} = T_{p:q} + T_{bruteforce} \tag{1}$$

where

$T_{setup}$  = Time taken to interrupt our setup.

$T_{p:q}$  = Time taken to search out  $p$  and  $q$  using Fermat factorization.

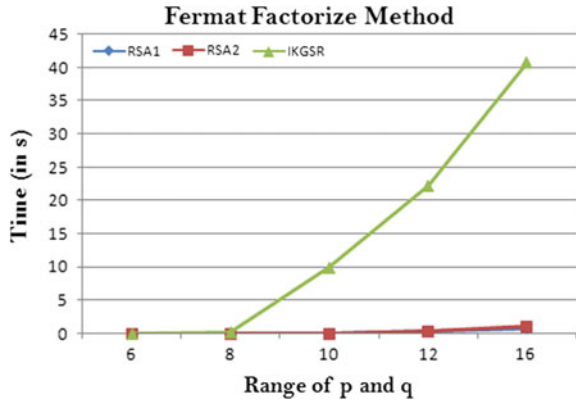
$T_{bruteforce}$  = Time taken for brute-force attack.

Factorization [13] is the reverse method of multiplication. It's an act of dividing a larger number into a group of smaller numbers called factors, which, when multiplied, form the original number. For a given large number, it is difficult to find the appropriate factorization used in a crypt method. In public key cryptography, an attacker tries to detect private key from public key, which is made up of the larger prime numbers. Factorization of such bigger numbers could be a problematic method and may take much time as shown in Fig. 5.

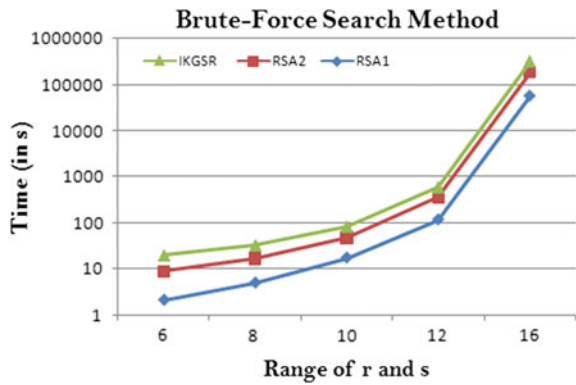
A brute-force attack involves making an attempt to find desirable key until an intelligible adaptation of the ciphertext into plaintext is obtained. On average, at least half of all desirable keys must be tried to achieve success, and results of brute-force attack is shown in Fig. 6.

After finding factorization time and brute-force search time, we can easily find out the total time to break proposed setup using the Eq. 1. The Figs. 5 and 6 clearly show that proposed framework has good security feature compared to other two techniques we considered for comparison. The speed of the proposed framework is higher as shown in Fig. 7 compared to traditional RSA algorithmic rule, because we are using indexes to fetch the value of public- and personal-key.

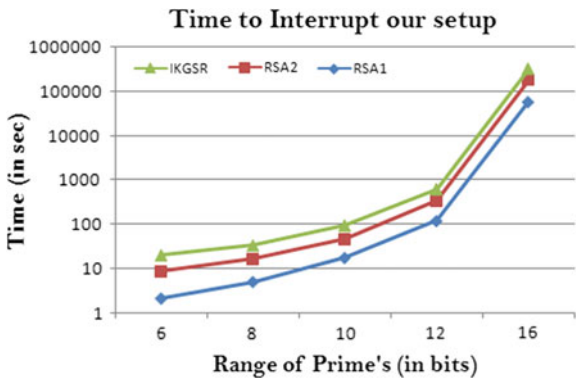
**Fig. 5** Time comparison for Fermat method



**Fig. 6** Time comparison for bruteforce method



**Fig. 7** Time comparison to interrupt total system



## 5 Conclusion

In this paper, (IKGSR) an improved key generation scheme using RSA algorithmic rule with disconnected cloud storage environments is proposed. In this proposed framework, since key parameters are kept offline since the system initiation, speed of proposed method is higher compared to other RSA methods considered. Additionally, usage of four prime numbers leads to additional time required to search these primes since the public- and personal-key computations are based on the values of  $Z$  (not  $n$ ). Hence the proposed model offers the improved security level and speed by using offline repository model.

## References

1. Mell, P.M., Grance, T.: SP 800-145, the NIST definition of cloud computing. Technical report, NIST, Gaithersburg, MD, United States (2011)
2. Buchade, A.R., Ingle, R.: Key management for cloud data storage: methods and comparisons. In: Fourth International Conference on Advanced Computing and Communications Technologies, pp. 263–270 (2014)
3. Ali, M., Khan, S.U., Vasilakos, A.V.: Security in cloud computing: opportunities and challenges. *Inf. Sci.* (2015). <https://doi.org/10.1016/j.ins.2015.01.025>
4. Stallings, W.: *Cryptography and Network Security: Principles and Practice*, 5th edn, p. 121e44, 253e97. Pearson Education (2011)
5. Jamekar, R.S., Joshi, G.S.: File encryption and decryption using secure RSA. *Int. J. Emerg. Sci. Eng.* **1**, 11–14 (2013)
6. Somani, N., Mangal, D.: An improved RSA cryptographic system. *Int. J. Comput. Appl.* **105**, 16 (2014)
7. Patidar, R., Bhartiya, R.: Modified RSA cryptosystem based on offline storage and prime number. In: IEEE International Conference on Computing Intelligence and Computing Research, pp. 1–6 (2013)
8. Abd, S.K., Al-Haddad, S.A.R., Hashim, F., Abdullah, A.: A review of cloud security based on cryptographic mechanisms. In: International Symposium on Biometrics and Security Technologies (ISBAST), pp. 106–111 (2014)
9. Song, N., Chen, Y.: Novel hyper-combined public key based cloud storage key management scheme. *China Commun.* **11**, 185–194 (2014)
10. Thangavel, M., Varalakshmi, P., Murali, M., Nithya, K.: An enhanced and secured RSA key generation scheme (ESRKGS). *J. Inf. Secur. Appl.* **20**, 3–10 (2015)
11. Luy, E., Karatas, Z.Y., Ergin, H.: Comment on an enhanced and secured RSA key generation scheme (ESRKGS). *J. Inf. Secur. Appl.* (2016)
12. Wagner, N.R.: The laws of cryptography with java code. Technical report, pp. 78–112 (2003)
13. Bishop, D.: Introduction to cryptography with java applets, pp. 237–250 (2003)

# Multi-QoS and Interference Concerned Reliable Routing in Military Information System



V. Vignesh and K. Premalatha

**Abstract** Secured and trustable routing in military information system is a sophisticated task in which sharing of information with no distortion or collusion is important. Mobile ad hoc networking enables the military communication by forwarding the information to the corresponding nodes on the right time. In the available system, Neighborhood-based Interference-Aware (NIA) routing is performed over an environment of stabilized position node. This research cannot offer support to the military communication system where the troops of soldiers might have to move to different positions. This issue is resolved in the new research strategy by implementing the new framework known as Multi-Objective concerned Reliable Routing in Dynamic MANET (MORR-MANET). In this technical work, optimal routing is performed with due consideration to different QoS parameters making use of pareto-optimal approach. Once the optimal route path is found, interference is prevented through the monitoring of the information regarding the neighborhood. Moreover, this research work is also related to the path breakage due to the resource unavailability or node mobility, as observed in this research work making use of Modified Go-Back-N ARQ technique. The whole research is realized and thereafter shown in the simulation environment, and it is revealed that the proposed research methodology provides a better result, in comparison with the previous research work NIA. The proposed technique indicates better performance in terms of 13% better residual energy, 7% larger the number of live nodes, 13% least relative error compared to the available technique NIA.

**Keywords** Military communication • Priority information • QoS satisfaction  
Bandwidth allocation

---

V. Vignesh (✉)

Department of IT, Sri Ramakrishna Engineering College,  
Coimbatore, TamilNadu, India  
e-mail: cbe.venkatvignesh@gmail.com

K. Premalatha

Department of CSE, Bannari Amman Institute of Technology,  
Sathyamangalam, TamilNadu, India  
e-mail: kpl\_barath@yahoo.co.in

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_32](https://doi.org/10.1007/978-981-10-7200-0_32)

351

## 1 Introduction

The military aspects found in a mobile ad hoc network offer a great deal of interest and are complex. A military scenario in addition to a hostile environment has many things to be considered and meticulous constraints rather than a MANET used for educational or business. For instance, a military scenario might possess greater necessities concerned with the information security. As the name implies, a mobile ad hoc network consists of mobile nodes. This, in turn, means that a node can transport itself physically between various locations in the network and be in and out of reach from the nodes present in the same network, owing to the range of radio propagation [1].

The performance of MANET in the military communication might be degraded in case of occurrence of interference. The interference in MANET can occur when the information is passed onto the different troops that share the same neighboring nodes. In a case like this, the information, which is transmitted, could be corrupted or left out; hence, the right information on the right time is not received by the troops of soldiers. This has to be prevented for performing a better transmission of information to the troops of soldiers who vary their locations dynamically. In this research work, this problem is solved by implementing the new framework called as Multi-Objective concerned Reliable Routing in Dynamic MANET (MORR-MANET).

The QoS routing issue in ad hoc networks has motivated the researchers in the recent times, and different solutions have been developed in order to guarantee end-to-end quality for flows. The initial solutions considered the bandwidth over an ad hoc link in isolation and attempted to find the paths, which satisfied the quality requirements (e.g., [2, 3]). Such type of solutions did not take the interference between the neighboring links, or that is seen between the different hops in the same flow into consideration. Open Shortest Path First (OSPF) routing protocol is basically an IETF specified link state routing protocol for the Internet [4]. The protocol extended for OSPF is QOSPF (Quality Of Service Open Shortest Path First) [5]. For the purpose of increasing the QoS in order to meet the requirement of real-time traffic, e.g., video conference, streaming video/audio, QoS-aware routing protocols are taken into account for Internet.

Zone Routing Protocol (ZRP) is actually a hybrid routing protocol [6]. It efficiently integrates the benefits of both proactive and reactive routing protocols. In Dynamic Load-Aware Routing protocol (DLAR), network load is defined to be the number of traffic ongoing in its interface queue [7]. The Load-Sensitive Routing (LSR) protocol defines the network load as seen in a node to be the sum of the number of packets that are queued in the interface of the mobile host along with its neighboring hosts [8]. Zafar et al. [9] introduced a novel capacity-constrained QoS-aware routing scheme known as the shortest multipath source: Q-SMS routing that lets the node to get and thereafter make use of the residual capacity estimation to carry out suitable admission control decisions.

Singh et al. [10] introduced the prediction of end-to-end packet delay seen in mobile ad hoc network using AODV, DSDV, and DSR on the basis of Generalized Regression Neural Network (GRNN) and radial basis function. But there exists no link recovery provisioning. Surjeet et al. [11] introduced a new on-demand QoS routing protocol MQAODV for the case of bandwidth-constrained delay-sensitive applications in MANETs. Wang et al. [12] introduced a protocol, which utilizes an alternative path solely if the data packets cannot be delivered through the primary route. Chia-Pang Chen et al. [13] introduced a hybrid genetic algorithm for improving the network life span of wireless sensor networks. Chuan-Kang Ting and Chien-Chih Liao [14] introduced a Memetic Algorithm (MA) for resolving the K-COVER issue during the WSN improvement. Ahmed E.A.A. Abdulla et al. [14] designed a hybrid approach that merges flat multihop routing and hierarchical multihop routing techniques. Behnam Behdani et al. [15] studied about the Mobile Sink Model (MSM) and queue-based delay-tolerant MSM.

## 2 Secured and Interference-Aware Military Communication

Military communication serves to be the most significant element during the times of war for commanding or sharing the information with troops of soldiers, located in different destinations or places. In the classical world, military communication is conducted by humans, which is not very much efficient due to the delay in the delivery of information. This research work is related to the path disturbances happening owing to the resource unavailability or node mobility.

- (1) Establishment of the route path based on QoS satisfaction level.
- (2) Locating the disrupted path and then rerouting the packets.
- (3) Locating the effect of interference dynamically with due concern given to the behavior of the nodes during mobility.

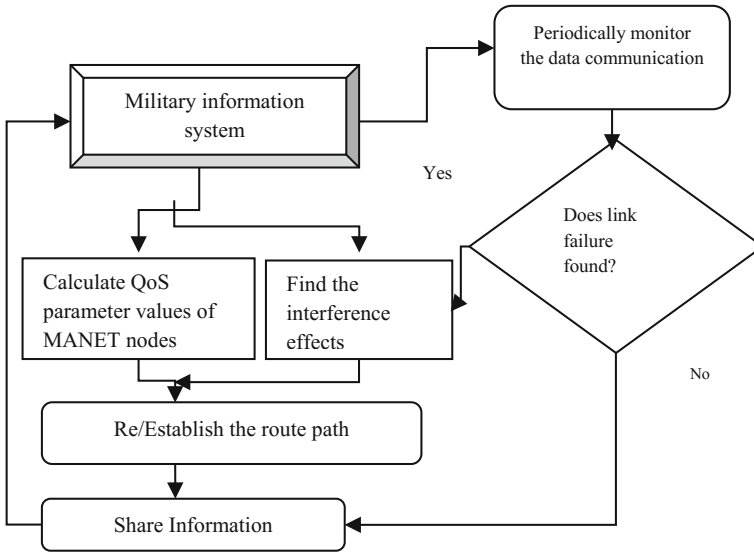
The steps mentioned above are performed in order to achieve a better routing in the MANET environment; hence, the military communication can be provisioned with security and effectiveness. The overall flow of the proposed research strategy is illustrated in Fig. 1.

The diagram shown above provides the detailed overview of the proposed research methodology, concerning the accomplishment of the secured and reliable information sharing with the troops of soldiers, positioned in different places. The detailed description of the newly introduced research methodology is given in the following subsections along with suitable examples and diagrams.

### A. QOS-aware Route Establishment in Military Communication Field

In the MANET environment, nodes are basically wireless devices containing only less resource. Here, the establishment of route with no consideration to the





**Fig. 1** Overall flow of the proposed research methodology

limitation in resource would result in routing failure and leads to the packet drop or delay. But in military communication, rapid and accurate delivery of information is the most necessary aspect that might decide the victory of war. Therefore, considering the QoS factors during the establishment of route path is the most necessary task. In this research, QoS factors taken into consideration are the available bandwidth, available power, end-to-end delay, and the stability of link employed for making the route selection. The node satisfying all these parameter values must be chosen for the establishment of route. In this research, the weighted sum technique is employed for the optimal route path selection, which meets the QoS parameter values.

- (1) **Available Bandwidth (BW)**: Available bandwidth (BW) indicates the available link bandwidth in the path from source node to the destination node multicast tree.
- (2) **Available Power (P)**: The available power of a node in multicast tree is represented in Eq. (1).

$$P = P_{\text{Total}} - E_{\text{consumed}} \quad (1)$$

where  $P_{\text{Total}}$  refers to the total energy at a node, and it is predetermined and fixed for every node present in the network.

- (3) **Available Delay (D)**: The delay (D) is defined as the maximum value of delay in the path from source node to destination nodes.

- (4) **Stability of Link:** In this research, a QoS-aware metric is proposed in order to decide over a stable link on the basis of the Link Stability Factor (LSF). The stability factor is estimated making use of contention count, received signal strength, and hop count in the form of QoS parameters.
- (5) **Reliability:** The degree to which the result of a measurement, calculation, or specification can be depended on to be accurate.

## B. Pareto-Optimal Method

Pareto-optimality is a concept seen in multi-criteria optimization, which permits for the optimizing a vector of multiple criteria, facilitating all the trade-offs observed among the optimal combinations of multiple criteria to be assessed. Pareto-optimality owes its origin to the concept of efficiency in economics and has been recently used for different issues in ecology. In the algorithm proposed, only the non-dominated solutions are stored. First, the population is sorted based on the decreasing order of significance to the first objective value. In this manner, the solutions that are good in first objective will arrive first in the list and those with bad value will be the last.

The reason behind can be described as below; this solution can be dominated only by the first solution (best in first objective); it cannot get dominated by other solutions just because its value for the first objective function is higher compared to the other solutions except first. In a similar manner, for the third solution, at most of two comparisons are needed from first and second points. And then for the final point of list, this solution has to be compared with every non-dominated solution. In case the solutions in list are not distinct in the first objective function value, then few changes have to be made in the algorithm proposed. It can be, checking each solution to its next immediate successors, and when any immediate solution dominates this solution, then this point has to be removed from the non-dominated set  $S1$ . At last, the non-dominated solutions are displayed. The algorithm proposed can be executed making use of the following steps.

- (1) Sort all the solutions ( $P1...PN$ ) in descending order of their first objective function ( $F1$ ) and generate a sorted list ( $O$ ).
- (2) Initialize a set  $S1$  and add the first element of list  $O$  to  $S1$ .
- (3) For each solution  $O_i$  (other than first solution) of list  $O$ , compare the solution  $O_i$  from the solutions of  $S1$ .
- (4) If any element of set  $S1$  dominate  $O_i$ , remove  $O_i$  from the list.
- (5) If  $O_i$  dominate any solution of the set  $S1$ , remove that solution from  $S1$ .
- (6) If  $O_i$  is non-dominated to set  $S1$ , then update set  $S1 = S1 \cup O_i$ .
- (7) If set  $S1$  becomes empty, add the immediate solution at immediate solution to  $S1$ .
- (8) Print non-dominated set  $S1$ .

## C. Path Breakage Detection to Avoid the Packet Loss/Delay

Path breakage plays a significant role in the military communication environment, where the information may be lost or else there will be a delay in delivery. Path breakage seen in the MANET environment needs monitoring for avoiding the

delayed delivery of information. In this work, it is performed by introducing the Go-Back-N ARQ method.

Go-Back-N ARQ is a particular instance of the Automatic Repeat reQuest (ARQ) protocol, in which the sending process continually sends a number of frames that are defined by a window size even with no receipt of an acknowledgment (ACK) packet obtained from the receiver. It is a certain case of the general sliding window protocol having the transmit window size of  $N$  and a receive window size of 1. It is capable of transmitting  $N$  frames to the peer before needing an ACK.

The receiver process maintains and keeps track of the sequence number of the next frame that it anticipates to receive, and then sends that number with each ACK it transmits. The receiver will leave out any frame, which does not possess the exact sequence number it is expecting (either a duplicate frame already acknowledged by it, or an out-of-order frame it is expecting to receive at a later point of time) and will then resend an ACK for the last correct in-order frame [1]. When the sender has sent every one of the frames in its window, it will find that all the frames till the first lost frame are outstanding and will look back to the sequence number of the last ACK it obtained from the receiver process and fills its window beginning with that frame and then continues the process over and again.

Go-Back-N ARQ uses a connection more efficiently than stop-and-wait ARQ, as here, dissimilar to waiting for an acknowledgment for every packet, the connection is still being used as the packets are getting sent. Otherwise said, the time, which would otherwise be spent in waiting, will be used for sending more packets. But this technique also causes the transmission of frames several number of times—when any frame was lost or corrupted, or the ACK that acknowledges them was lost or corrupted, then that frame and all of the next following frames in the window (even if they were received with no error) will get resent. In order to prevent this, selective repeat ARQ can be utilized.

There are certain things to consider while selecting a value for  $N$ :

- (1) The sender should not transmit too rapidly.  $N$  must be bounded by the receiver's capability of processing the packets.
- (2)  $N$  should be smaller compared to the number of sequence numbers (in case they are numbered from zero to  $N$ ) in order to verify the transmission in situations of any packet (any data or ACK packet) getting dropped.
- (3) With the bounds given in (1) and (2), select  $N$  to be the biggest number possible.

#### **D. Inference Avoidance using Neighborhood Information and the Path Breakage Information**

While the data is transmitted, a node might receive two or more similar packets, leading to interference and redundancy. For a particular node in the network, just the neighbors who send or forward the packets (i.e., the active neighbors) will be interfering with it. Therefore, the other neighborhood nodes will not be affected by it. The interference index of a path is defined as the sum of interference index values of

the component links. Hence, in a single channel Time Division Duplex (TDD) network in MANETs, any broadcast transmission adopts the principle that there should be just one node that can do the transmission among the neighbors of a receiver. In addition, every mobile device cannot simultaneously act as a sender and receiver.

#### **E. Minimizing Interference using the Node with Fewer Neighbors**

With the intent of developing of a more interference-effective variant of GBR-CNR, the number of neighbors is considered in the receiving node. The concept here is that reducing the number of neighbors that surround the receiving node will reduce the chances that there shall be a receiver's neighborhood node also functioning as a transmitting node in the same time slot just as the sender. Suppose the nodes labeled A are senders, nodes B are neighbors of A, and node D is the destination. Node A will select B2 as a next hop, rather than B1 as the numbers of neighbors of B2 are lesser compared to the neighbors of B1. Lesser neighbors can be considered as lesser chances of corrupted packets; therefore, in turn, an increase in network throughput can be attained.

#### **F. Minimizing Interference using the Node that is Less Used**

Examining another variation of the approach mentioned above for achieving more interference-efficient routing making use of GBR-CNR, the number of communications the receiving node is already taking part in is taken into consideration. The algorithm is GBR-CNR with the less utilized (GBR-CNR-LU) nodes selected to be next hops. Consider that the nodes that are labeled A are senders, nodes B are the neighbors of A, and node D acts the destination. It is assumed that there exists two paths and node B1 is selected to be the next hop for node A1. Till now, when the protocol will be establishing the second path, node A2 will choose, for the next hop, node B2 in place of B1 as node B1 takes part in most communications compared to node B2 although node B1 is nearer to the destination D2 rather than node B2. Therefore, a node participating in lesser number of communication paths is less vulnerable to message degradation.

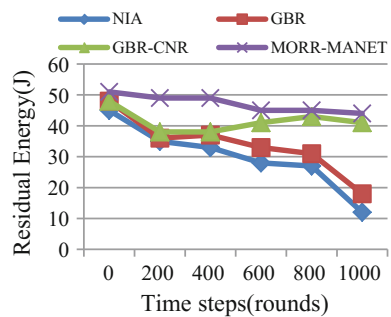
### **3 Experimental Results**

This section studies about the performance of the proposed scheme, which is assessed through network simulation. Here, the comparison is performed between the new research called as the MORR-MANET and the already available technical work known as neighborhood-based interference avoidance (NIA), Greedy-based Backup Routing protocol (GBR), and GBR using Conservative Neighborhood Range (GBR-CNR). The performance measurement is done in terms of residual energy, number of nodes alive, and relative error. The setting values used for the network configuration when the experiments are carried out are given in Table 1. These values can be varied and optimized for different applications. In the present

**Table 1** Setting values for experiments

Parameter	Value	Unit	Description
N	30	Nodes	Total number of nodes
C	3	Clusters	Number of clusters
$T_{recluster}$	30000	Ms	Time to recluster
$T_{sample}$	50	Ms	Sample time for sensing
$T_{cycle}$	5000	Ms	Time interval between two data transmission
$T_{DataRx}$	500	Ms	Time to receive data of CH
$T_{Dataagg}$	50	Ms	Time to aggregate data at CH
$T_{Radioon\_CH}$	600	Ms	Maximum time to keep radio on for sending
$T_{Radioon\_CM}$	100	Ms	Maximum time to keep radio on for sending
$\Delta V_{th}$	100	mV	Voltage threshold for dead node

**Fig. 2** Total remaining energy of the network



case of study, the values of the time intervals in Table 1 are chosen such as to minimize the experimental time duration for the purpose of observation.

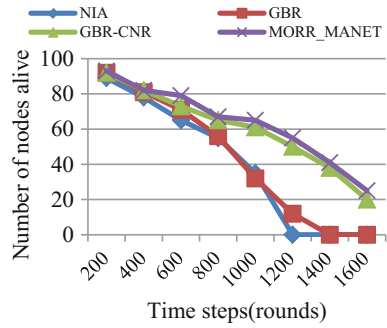
The performance measure values that are assessed for comparison and illustrating the improvement of the research methodology introduced is shown in Figs. 2, 3 and 4.

Figure 2 illustrates the total amount of the remaining energy in the network. It is evident from the figure that the energy consumed by the proposed scheme is significantly smaller in comparison with the others, particularly in the first rounds. This is due to an effective cluster formation. The proposed technique is 13% better compared to the already available techniques.

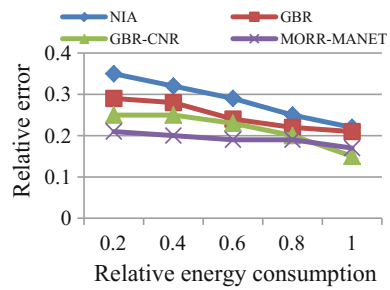
Figure 3 shows the number of live sensor nodes with the round starting with 0.5 J/node at the beginning. It reveals that the time when the first node dies with the new MORR-MANET technique is about 7% higher in comparison with that of NIA. The time when every node dies with the proposed scheme is also significantly greater compared with them.

Figure 4 illustrates above shows the performance comparison of the energy consumption methods vs relative error as measured in the novel MORR-MANET compared against the existing NIA. Similar to the results, it reveals that the

**Fig. 3** Comparison of the number of live nodes as the round proceeds



**Fig. 4** Reconstruction accuracy and energy consumption



proposed MORR-MANET yields 13% least relative error for a certain energy expenditure compared to the other available techniques.

### 4 Conclusion

Military communication is a problem of importance remaining in highlight for various research works where the requirements of routing have to be achieved, targeting at the accomplishment of reliable and secured data communication. In this research work, a novel framework known as MORR-MANET is presented, which focuses over several factors such as QoS parameters, path breakage consideration along with the impacts of interference. Hence, the reliable and secured information passing is enabled in the military communication system. The system introduced assures rapid and secure communication of the information to the troop of soldiers present in the environment. Hence, reliable and secured communication can be achieved in military. The results from the experiments indicate that the research methodology proposed aids in yielding the better result in comparison with the existing research.

## References

1. IHS Jane's Military Communications Retrieved 2012-01-23
2. Bollobás, B.: *Modern graph theory* Springer Science & Business Media, vol. 184 (2013)
3. Zhang, X.M., Zhang, Y., Yan, F., Vasilakos, A.V.: Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Trans. Mob. Comput.* **14**(4), 742–754 (2015)
4. Sultan, N.T., Jamieson, D.D., Simpson, V.A.: U.S. Patent No. 7, 831,733. Washington, DC: U.S. Patent and Trademark Office (2010)
5. Ahn, C.W., Ramakrishna, R.S.: A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans Evol Comput.* vol. 6, no. 6, pp. 566–579 (2002)
6. Jamwal, D., Sharma, K.K., Chauhan, S.: *Zone Routing Protocol* (2014)
7. Lee, S.J., Gerla, M.: Dynamic load-aware routing in ad hoc networks. In: *International Conference on Communications*, vol. 10, pp. 3206–3210 (2001)
8. Rexford, J.L., Shaikh, A.: U.S. Patent No. 6, 801, 502. Washington, DC: U.S. Patent and Trademark Office (2004)
9. Zafar, H.: QoS-aware Multipath Routing Scheme for Mobile Ad Hoc Networks. *Int J Commun Netw Inf Secur* **4**, 1–10 (2012)
10. Singh, J.P.: Delay prediction in mobile ad hoc network using artificial neural network. *Proced. Technol.* **4**, 201–206 (2012)
11. Surjeet, B.: QoS bandwidth estimation scheme for delay sensitive applications in MANETs. *J. Commun. Netw. Sci. Res.* **5**, 1–8 (2013)
12. Wang, J.: QoS routing with mobility prediction in MANET. In: *Proceedings of the IEEE Pacific Rim Conference on Computers and Signal Processing*, Victoria, BC, Canada, pp. 357–360 (2001)
13. Ting, C.K., Liao, C.C.: A memetic algorithm for extending wireless sensor network lifetime. *Inf. Sci.* **180**(24), 4818–4833 (2010)
14. Abdulla, A.E., Nishiyama, H., Kato, N.: Extending the lifetime of wireless sensor networks: a hybrid routing algorithm. *Comput. Commun.* **35**(9), 1056–1063 (2012)
15. Behdani, B., Yun, Y.S., Smith, J.C., Xia, Y.: Decomposition algorithms for maximizing the lifetime of wireless sensor networks with mobile sinks. *Comput. Oper. Res.* **39**(5), 1054–1061 (2012)

# A Secure Cloud Data Storage Combining DNA Structure and Multi-aspect Time-Integrated Cut-off Potential



R. Pragaladan and S. Sathappan

**Abstract** Confidentiality and authentication enable cloud storage server to prove that it is storing owner's data honestly. However, most of the constructions suffer from special update techniques and the issue of a complex data modification, which might hinder the deployment of confidentiality and authentication in practice. In this regard, we propose a DNA-based Multi-aspect Cut-off Potential (DNA-MACP) framework, by making use of DNA-based Watson–Crick–Hoogsteen and Time-Integrated Cut-off Potential to reduce the time complexity for establishing data confidentiality and space complexity by managing one-time password. Based on the intractability of the tertiary triples, a secure confidential data transaction is designed for cloud storage, where the user authentication scheme is utilised to deal with the improper data modification problem. The DNA-MACP framework enhances the authentication level of security by using both confidentiality and authentication techniques from the unauthorised user modification. The DNA-MACP framework with extensive security analysis and implementation results in reducing both time and space complexities in the business data transactions in a cloud environment.

**Keywords** Confidentiality · Authentication · DNA · Multi-aspect Cut-off potential · Watson–Crick–Hoogsteen

---

R. Pragaladan (✉) · S. Sathappan  
PG and Research Department of Computer Science, Erode Arts and Science College,  
Erode, Tamil Nadu, India  
e-mail: pragaladanr@gmail.com

S. Sathappan  
e-mail: devisathappan@yahoo.co.in



## 1 Introduction

Migration of sensitive information, managing confidential data in the cloud, and intensive measures to securitize the virtualised environment in the cloud provides practical solutions. Hence, data confidentiality and authentication of cloud users play a significant role and have a meaningful impact towards ensuring security to cloud data storage in the business transaction processing.

Cloud computing is an Internet-based computing which shares resources, utilities, software and information to other devices on demand basis from one system to another, and their increased growth rate has resulted in the substantial research and development area. Here, we consider combining transactional data confidentiality and cloud user authentication to support secure and high-performance storage systems in cloud environment. The main aim is to design and develop a data confidentiality mechanism to ensure privacy with the transactional data stored in cloud server and improves data confidentiality in maximally. To provide authentication, the authorised cloud user accesses the confidential transactional data stored in cloud server such that the time and space complexity minimised.

This paper describes the drawbacks of the existing methods and their related works in Chap. [An Ontology-Based Approach for Automatic Cloud Service Monitoring and Management](#). Chapter [Incorporating Collaborative Tagging in Social Recommender Systems](#) defines the proposed framework to the system models. Chapter [Twitter Sentimental Analysis on Fan Engagement](#) offers the experimental setup, and the discussion is explained in Chap. [A Hybrid Semantic Algorithm for Web Image Retrieval Incorporating Ontology Classification and User-Driven Query Expansion](#). The paper is concluded in Chap. [Attribute Selection Based on Correlation Analysis](#).

## 2 Related Work

Pragaladan et al. [1] used the method called Watson–Crick–Hoogsteen-based data confidentiality (WHO-CDT) structure by using a 2-bit binary form of DNA structure to enhance data privacy by using cloud business data transactions. Data confidentiality is of particular importance because of the significance it possesses. To consider medical information is of immense importance that maintains the patient history and has to be protected. Such a need has motivated different types of methods aiming at ensuring both data confidentiality and data ownership. This approach is shown in the Suppressed-based and Generalised-based Update (SGU) developed by Alberto Trombetta et al. [2], and Cloud-based Virtual Phone (CVP) model by Jiun Hung Dinga et al. [3] enabled the users to provide a secure and wistful environment with enhanced performance and protected privacy.

Pragaladan et al. [4] describe the method called Fast Chaos-based DNA Cryptography (FSB-DC) by using a 3-bit binary form of DNA structure with fast chaos-based random number generation to improve data confidentiality by using cloud business data transactions. This method succeeds the DNA-based

cryptography which ensures secure data storage on multi-clouds (DNA-CMCS) developed by Ranalkarn R. H. et al. [5], and Siddaramappa V. [6] describes a technique using DNA sequence with numbers in Random Function and Binary Arithmetic Operations (RFBAOs). The numbers are random numbers which are used to provide high data security.

Pragaladan et al. [7] proposed the multi-aspect Sparse Time-Integrated Cut-off Authentication (STI-CA) method by using a multi-aspect sparse one-time password to progress data confidentiality into maximum. This framework accomplishes the Wenjuan Xu et al. [8] method of Dynamic Remote Attestation Framework and Tactics (DR@FT) which adopts a graph-based method to represent integrity violations and the graph-based policy analysis in cloud infrastructure. Kurt Oestreicher [9] introduced iCloud service via native Mac OS X system (iCloud native Mac OS X) used by file hash values to match the original files using the MD5 algorithm. Synchronisation integrity schema research was not carried out to establish the authenticity of cloud data-based transactions.

Bogdan Carbutar et al. [10] introduced a framework for providing security and privacy protection for integrating the cloud and volunteer computing paradigms which ensure security for outsourced data and service providers in distributed applications. Jucheng Yang et al. [11] hosted a fingerprint recognition system using the geometric and Zernike moment to ensure secure communication between cloud users and cloud server. A comprehensive survey for deploying the data-intensive application in the cloud was presented in Sherif Sakr et al. [12] model. Data object replication was designed in Manghui Tu et al. [13] scheme to minimise the communication cost involved in data transfer. Despite authenticity, the communication and computation cost remained a major issue to be addressed. Zhuo Hao et al. [14] developed a protocol called a remote data integrity checking which was designed for cloud storage. This protocol also proved to be efficient in the aspects of communication, computation and storage costs. With the rise of big data in cloud computing, a review of work with a focus on scalability, data integrity and data transformation was provided in Ibrahim Abaker et al. [15] model. A comprehensive study of authentication survey methods was presented in Mojtaba Alizadeh et al. [16] to access cloud-based resources.

Another improved user authentication system was designed in SK Hafizul Islam et al. [14] by applying enhanced two-factor authentication system ensuring security. Protecting the privacy of metadata was considered in Yves-Alexandre de Montjoye et al. [15] structure through anonymisation and dimensionality of data shared. However, the cost involved increased with the increase in the dimensionality of data. To eliminate this issue, Muhammad Shiraz et al. [16] developed a model using distributed computational offloading to reduce the resource utilisation as a lightweight structure.

With a recent focus on cloud computing, identity management is one of the critical issues for the viability of any cloud environment. This domain has also experienced appreciable regard not only from the research community but also in the IT sector. Several Cloud Identity Management Systems (IDMSs) have designed so far. Umme Habiba et al. [17] presented a taxonomy of security issues and solutions towards IDMS. Hyungjoo Kim et al. [18] introduced a method used to ensure the safety based on zero knowledge authentication protocol designed in a

multi-cloud environment. A new direction in cloud computing security using cryptography mechanism was developed in Mehmet Sabır Kiraz et al. [19] model. A secure authentication scheme using Chebyshev chaotic maps was conceived in Suye Namasudra et al. [20] system.

Based on the above-mentioned methods and techniques, here we describe a work called, a DNA-based Multi-aspect Cut-off Potential (DNA-MACP) framework which is developed to ascertain the secure cloud data storage in the cloud infrastructure.

### 3 DNA-Based Multi-aspect Cut-off Potential (DNA-MACP)

In a cloud environment, cloud owners store their data remotely which are then used by authorised cloud users, resulting in higher quality of services on several cloud applications. In this paper, we introduce a new structure called DNA-based Multi-aspect Cut-off Potential (DNA-MACP) to ensure confidentiality of transaction data and user authentication in cloud storage environment.

This DNA-MACP framework includes cloud owners ' $CO_i = CO_1, CO_2, \dots, CO_n$ ', cloud storage servers ' $CSS$ ' and cloud users ' $CU_i = CU_1, CU_2, \dots, CU_n$ '. The cloud owners stored their transactional data ' $TD$ ' in a more confidential manner, and with the confidential transaction data, only authenticated users are acceptable to access the confidential transaction data from cloud storage server. The cloud storage servers ensure that the authorised cloud users only access the data.

The DNA-MACP framework required for cloud storage data needs to be more confidential and secure. This work aims to achieve confidentiality of cloud data using DNA structure-based confidentiality form. The DNA privacy structure type is the technique in which the user provided transactional data converted into binary storage stage by fragmenting the whole transactional data into three bits.

Data confidentiality is achieved by applying Tertiary Triplet Rule and then processing them by subjecting it to 3-bit DNA digital coding. Followed by authorised users are permissible to access the confidential transaction data with the help of Cut-off Potential. The architecture of DNA-based Multi-aspect Cut-off Potential framework that describes the flow to ensure data confidentiality and cloud user authentication is demonstrated in Fig. 1.

This DNA-MACP framework stores the cloud owner data in a secure and confidential manner. The cloud owner data stored in Ultra Compact Transactional Information Storage the model based on DNA, such as, ' $TD_1$ ', ' $TD_2$ ', ' $TD_3$ ', ..., ' $TD_n$ ' respectively and placed in Cloud Storage Server ' $CSS$ '.

This DNA-MACP framework comprises of the 3-bit Tertiary Triple Confidential Data Transaction (TTT-CDT) and the Time-Integrated Cut-off Authentication (TI-CA) for cloud data storage. In the 3-bit Tertiary Triple Confidential Data Transaction (TTT-CDT), the Oligonucleotide Sequences-based Data Encoding

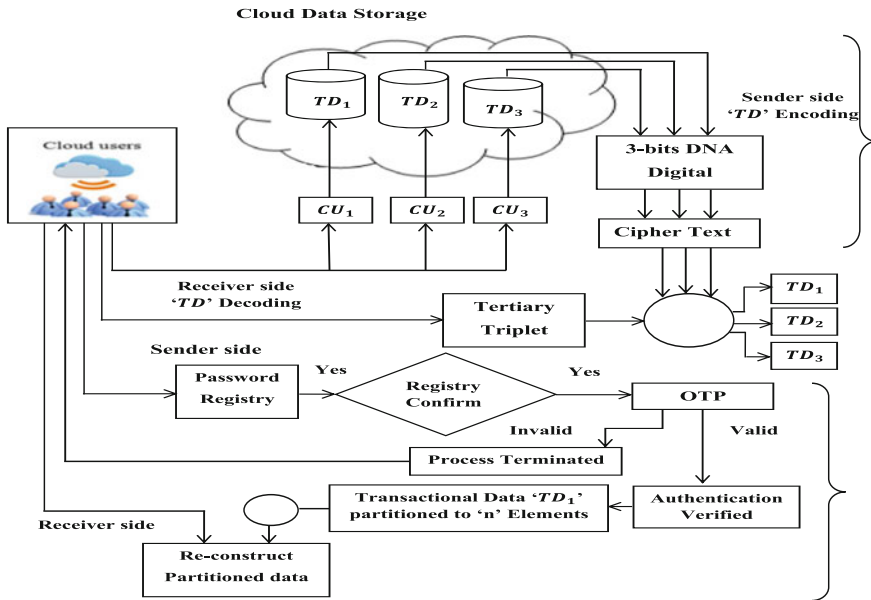


Fig. 1 Architecture of the DNA-based Multi-aspect Cut-off potential structure

Algorithm is used for 3-bit DNA digital coding transactional data that is stored efficiently using the mechanism. The transactional data are encoded using the Oligonucleotide Sequences-based Data Encoding Algorithm, with the cipher text stored in cloud storage server. On the other end, decoding is performed on the receiver side by applying Tertiary Triples to achieve a higher level of confidentiality and protect the transactional data in Atito et al. [5] method. The unique features of 3-bit Tertiary Triple include orders of magnitude transactional data encoding performed with minimum two 1-bit occupied position ('011', '101', '110' and '111'), whereas the values of other positional bits are remained empty ('000', '001', '010' and '100').

The Time-Integrated Cut-off Authentication (TI-CA) is used to perform user authentication whenever the cloud user has to access the confidential transactional data stored in cloud storage server. It is achieved by using the Cut-off Potential Cryptography mechanism. This Cut-off Potential Cryptography mechanism is deployed using a Time-Integrated One-Time Password (TI-OTP) algorithm. It is a two-layered encryption device used for implementing both the authentication and security for cloud data storage. The first layer performs the one-time password authentication, and the second layer prevents the unauthorised user modification on transactional data.

### 3.1 Three-Bit Tertiary Triple Confidential Data Transaction (TTT-CDT)

To provide confidentiality, the transactional data stored in cloud storage server for the DNA-MACP framework apply 3-bit Tertiary Triplets. Initially, the transactional data are stored in DNA structure. The four types of chemical bases found in DNA are adenine 'A', thymine 'T', cytosine 'C' and guanine 'G'. Table 1 shows the 3-bit DNA digital coding and their corresponding tertiary value.

The 3-bit Tertiary Triplets work on the principle that the DNA bases are stored with the aid of tertiary value which occupies the position with more than one 'one bit'. On the other hand, the other places are filled with '1', '2' and '3', respectively. In this way, a combination of ' $8^3 = 40320$ ' possible patterns is said to occur.

The advantage of using 3-bit DNA digital coding is that the tertiary and its corresponding DNA base value are not static, but keep on changing every time a cloud owner has to store transactional data in cloud storage server. Therefore, the confidentiality of the transaction data of each cloud owner is improved significantly. Oligonucleotide sequences are used to encode or decode the transactional data with minimum execution time, therefore improving the transactional data confidentiality level. Oligonucleotide Sequences-based Data Encoding or Decoding initially accepts the transactional data ' $TD_i = TD_1, TD_2, \dots, TD_n$ ' as input from different cloud owners ' $CO_i = CO_1, CO_2, \dots, CO_n$ ' to store them in the cloud storage server 'CSS'. With the obtained transactional data, the equivalent ASCII value is obtained and formulated as follows:

$$RES = ASCII(TD_i) \quad (1)$$

To the ASCII value, binary bits are formed and formulated as follows:

$$RES1 = BIT(RES) \quad (2)$$

The following algorithm shows the 3-bit tertiary triples mechanism.

---

**Input:** Transactional Data ' $TD_i = TD_1, TD_2, \dots, TD_n$ ', Cloud Owner ' $CO_i = CO_1, CO_2, \dots, CO_n$ ', Cloud Storage Server 'CSS',

---

**Output:** Optimal Time Complexity

1: Begin

**Transactional Data Encoding**

2: For each Transactional Data ' $TD_i$ '

3: Obtain ASCII ( $TD_i$ ) and store it in RES

4: Convert ASCII ( $TD_i$ ) into binary bits and store it in RES1

5 Form Three-bit Tertiary Triplets and Substitute their equivalent Oligonucleotide Sequences ' $OS_i$ ' (from table 1)

6: End for

**Transactional Data Decoding**

7: For each Oligonucleotide Sequences ' $OS_i$ '

8: Obtain Three-bit Tertiary Triplets

9: Perform ASCII ( $TD_i$ )

10: Extract Transactional Data ' $TD_i$ '

11: End for

12: End

---

**Algorithm 1: The procedure for Three-bit Tertiary Triplets mechanism**

**Table 1** 3-bit DNA digital coding

DNA base	0	1	2	3	T	A	G	C
Tertiary value	001	001	010	011	100	101	110	111

In the above algorithm, the new resultant value ‘RES1’, Oligonucleotide Sequences are formed from Table 1, where the resultant value forms the encoded transactional data. In a similar manner, the reverse operation is performed to extract the original transactional data. By combining the transactional data into its equivalent Oligonucleotide sequences and encoding us implement an efficient confidentiality mechanism is used to store transactional data in cloud storage server. The 3-bit Tertiary Triplets mechanism stores the transactional data, which obtained from the cloud owners of various data centres. The 3-bit Tertiary Triplet function is used to retrieve the transaction data from the cloud storage server by the corresponding cloud user.

### 3.2 Time-Integrated Cut-off Authentication (TI-CA)

Once the cloud owner’s transactional data is said to be confidential, the data access is made only by the authenticated cloud users. The DNA-MACP framework uses TI-CA to perform the cloud user authentication. The Time-Integrated Cut-off Authentication (TI-CA) mechanism comprises of two parts, and both are implemented using two-dimensional service matrices. An algorithm is designed to ensure authentication to the cloud users, using the Cut-off Potential Lagrange Coefficient Cryptography so that only the authorised cloud user accesses the confidential transaction data stored in cloud storage server.

In the first part, the time-integrated one-time password is applied to appropriate cloud user authentication continues until all the cloud users are either authenticated or till the process terminates with the unauthorised cloud users in the training process. In the second part, Cut-off Potential Cryptography mechanism is applied for authenticated cloud users where the transactional data of the corresponding cloud users are split into polynomial form and reconstructed using Lagrange coefficient. The reconstruction of transactional data is performed, and the training process continues until all the confidential transactional data is provided to the authenticated cloud users by cloud storage server. Algorithm 2 explains the procedure of Time-Integrated Cut-off Authentication for establishing the authenticity and also avoiding improper data modification.

---

**Input:** Cloud Service Provider ( $CSP_i = CSP_1, CSP_2, \dots, CSP_n$ ), Cloud Storage Server (CSS), Cloud Owner ( $CO_i = CO_1, CO_2, \dots, CO_n$ ), Cloud User ( $CU_i = CU_1, CU_2, \dots, CU_n$ ), Transactional Data 'TD', Time 't', elements 'c', total elements 'n'

---

**Output:** Optimal Space Complexity

---

```

1: Begin
2: For each Cloud Owner 'COi' and Cloud User 'CUi'
3: Store Uname,pwd in password registry 'PR'
4: If match occurs
5: Registry is said to be confirmed
6: Cloud Storage Server CSS sends OTP
7: For each time interval 't'
8: If OTP Authentication verified
9: For each Transactional Data 'TD'
10: Construct polynomial to divide transactional data
11: Obtain the values for 'c' and 'n'
12: For each 'c' values
13: Perform Lagrange Coefficient
14: End for
15: End for
16: End if
17: End for
18: End if
19: End for
20: If match does not occur
21: Process terminated
22: End if
23: End for
24: End

```

---

### Algorithm 2: The Cut-off Potential Lagrange Coefficient Cryptography

The algorithm illustrated three parts included in Cut-off Potential Lagrange Coefficient Cryptography. In the first part, username and password are stored in the registry, involving fan-out points which do not have any process. The second part performs the one-time password authentication on time-integrated basis. The drudgery of maintaining the password by cloud storage server is solved, resolving the time and computational complexity, by providing a one-time password to each cloud user, in a time-integrated manner. This part continues until the entire password registry verified cloud users given a one-time password to access the confidential transactional data placed in the cloud storage server. Finally, the third part uses the Lagrange coefficient for each partitioned transactional data that also prevents the unauthorised access.

## 4 Experimental Setup

The DNA-based Multi-aspect Cut-off Potential (DNA-MACP) framework on cloud environment is developed to improve the transactional data confidentiality and cloud user authentication for accessing the transactional data using the Amazon Access Sample, Amazon EC2 and Amazon Simple Storage Service (Amazon S3) data sets.

The JAVA coding is used to perform the experiment to achieve the confidentiality and authentication with CloudSim 3 platform quickly, which identifies the data privacy and user authentication before implementing in the real-world

scenario. The DNA-MACP framework simulated from the transactional data sizes varies from 10 to 70 KB on a cloud environment, and the experiment is conducted on factors such as data confidentiality, execution time, communication overhead and space complexity and ensures the level of authentication.

## 5 Result Analysis

The DNA-based Multi-aspect Cut-off Potential (DNA-MACP) framework on cloud environment is compared to the following existing methods with different data sets using business transactions.

### (A) DNA-MACP framework with Amazon Access Sample set

Using Amazon Access Sample Set, the DNA-MACP compared with WHO-CDT [1], SGU [2] and CVP [3]. The performance comparison of various parameters using Amazon Access Sample set is shown in Table 2:

- The data confidentiality improved by 27% compared to WHO-CDT, 36% compared to SGU and 46% compared to CVP. The execution time reduced 8% about WHO-CDT, 20% compared to SGU and 25% compared to CVP. The communication overhead reduced by 41% about WHO-CDT, 47% compared to SGU and 51% compared to CVP. The space complexity reduced by 14% about WHO-CDT, 26% compared to SGU and 33% compared to CVP.

### (B) DNA-MACP framework with Amazon EC2 data set

Using Amazon EC2 data Set, the DNA-MACP is compared with existing FSB-DC [4], DNA-CMCS [5] and RFBAO [6]. The performance comparison of various parameters using Amazon EC2 data set is shown in Table 3:

- The data confidentiality improved by 28% compared to FSB-DC, 43% compared to DNA-CMCA and 63% about RFBAO. The execution time reduced by 9% about FSB-DC, 27% compared to DNA-CMCA and 30% about RFBAO. The communication overhead reduced by 40% about FSB-DC, 47% compared to DNA-CMCA and 50% about RFBAO. The space complexity reduced by 24% about FSB-DC, 36% compared to DNA-CMCA and 40% about RFBAO.

### (C) DNA-MACP framework with Amazon S3 data set

Using Amazon S3 data set, the DNA-MACP compared with existing STI-CA [7], DRAFT [8] and iCloud native Mac OS X [9]. The performance comparison of various parameters using Amazon S3 data set is shown in Table 4:

- The data confidentiality improved by 21% compared to STI-CA, 42% compared to DRAFT and 62% about iCloud Native Mac OS X. The execution time reduced by 3% about STI-CA, 18% compared to DRAFT and 30% about iCloud Native Mac OS X. The communication overhead reduced by 18% about



**Table 2** Performance comparison of various parameters using Amazon Access Sample set

Transactional data size (KB)		Data confidentiality (KB)				Execution time (ms)			
		DNA-MACP	WHO-CDT	SGU	CVP	DNA-MACP	WHO-CDT	SGU	CVP
10		18.5	9.3	9.1	8.7	248	285	338	395
20		27.6	17.5	16.3	15.8	292	315	445	487
30		35.2	28.2	25.4	21.7	438	482	513	529
40		49.4	37.4	35.2	31.9	470	515	629	645
50		52.7	48.1	45.6	41.2	541	585	645	698
60		67.6	55.1	51.3	48.9	568	615	689	728
70		79.1	65.3	60.2	57.2	621	648	715	745
Transactional data size (KB)		Communication overhead (bps)				Space complexity (bps)			
		DNA-MACP	WHO-CDT	SGU	CVP	DNA-MACP	WHO-CDT	SGU	CVP
10		2.36	3.69	4.21	4.52	7	10	14	18
20		3.12	5.13	6.24	6.65	16	19	26	31
30		5.36	8.24	9.39	9.98	23	25	31	36
40		6.74	11.36	12.59	13.35	29	34	38	41
50		8.12	14.89	16.05	16.98	38	45	50	54
60		9.87	17.39	19.32	21.32	45	52	60	66
70		11.24	19.35	21.47	23.22	54	61	66	71

**Table 3** Performance comparison of various parameters using Amazon EC2 data set

Transactional data size (KB)	Data confidentiality (KB)				Execution time (ms)			
	DNA-MACP	FSB-DC	DNA-CMCA	RFBAO	DNA-MACP	FSB-DC	DNA-CMCA	RFBAO
10	19.5	11.3	7.6	7.1	244	276	312	325
20	30.5	19.5	15.5	14.5	279	308	486	495
30	40.1	29.4	23.2	21	412	450	522	540
40	50.5	38.3	33.7	29.5	438	502	624	650
50	57.9	50.5	45.4	38.6	506	565	693	723
60	71.5	57.7	56.2	51.2	542	602	712	765
70	80.2	67.3	62.8	52	600	628	789	799
Transactional data size (KB)	Communication overhead (bps)				Space complexity (bps)			
	DNA-MACP	FSB-DC	DNA-CMCA	RFBAO	DNA-MACP	FSB-DC	DNA-CMCA	RFBAO
10	1.9	3.57	4.84	5.61	6	12	17	19
20	2.8	4.89	5.65	6.57	14	21	29	33
30	5.3	8.04	8.71	9.81	20	27	35	37
40	6.4	11.18	13.43	14.51	27	35	41	44
50	7.6	13.42	15.52	16.25	35	48	52	56
60	9.3	14.89	16.02	16.85	43	55	65	67
70	10.6	16.72	18.92	19.2	52	62	70	75

**Table 4** Performance comparison of various parameters using Amazon S3 data set

Transactional data size (KB)	Data confidentiality (KB)				Execution time (ms)			
	DNA-MACP	STI-CA	DRAFT	iCloud Native Mac OS X	DNA-MACP	STI-CA	DRAFT	iCloud Native Mac OS X
10	20.4	14.2	11.3	7.6	240	256	296	312
20	33.3	22.3	19.5	15.5	267	302	345	486
30	44.7	33.7	27.4	23.2	390	398	450	522
40	51.6	45.4	36.3	31.4	406	412	532	624
50	63.2	56.2	46.5	42.1	474	482	596	693
60	76.5	62.8	53.7	51.1	518	521	602	712
70	82.4	72.7	67.3	59.3	581	592	688	789
<i>Transactional data size</i>								
Transactional data size (KB)	Communication overhead (bps)				Space complexity (bps)			
	DNA-MACP	STI-CA	DRAFT	iCloud Native Mac OS X	DNA-MACP	STI-CA	DRAFT	iCloud Native Mac OS X
10	1.59	2.89	3.67	4.72	5	9	12	17
20	2.56	3.42	5.22	6.13	13	17	23	29
30	5.24	5.13	8.24	9.72	18	24	27	35
40	6.05	7.24	11.36	13.36	25	32	35	41
50	7.12	8.45	13.42	15.72	32	41	48	52
60	8.67	10.87	14.89	16.39	41	48	55	65
70	10.14	12.24	16.72	18.35	50	59	62	70

STI-CA, 44% compared to DRAFT and 51% about iCloud Native Mac OS X. The space complexity reduced by 20% about STI-CA, 30% compared to DRAFT and 40% about iCloud Native Mac OS X

## 6 Conclusion

In this paper, DNA-based Multi-aspect Cut-off Potential (DNA-MACP) framework is provided based on the 3-bit Tertiary Triplets mechanism and Cut-off Potential Lagrange Coefficient Cryptography algorithm for executing different cloud user requests with various transactional data sizes. This method improves the data confidentiality stored in cloud storage server and avoids the access of unauthorised user in a remote manner as the framework uses 3-bit Tertiary Triplets work in a dynamic structure used to reduce the time complexity involved in accessing the confidential transactional data in a cloud environment. By applying the Cut-off Potential Cryptography mechanism in DNA-MACP framework, it improves the space complexity utilising the one-time password for each cloud user, reducing the drudgery of cloud storage server in storing the password. It reduces the space complexity between successive cloud user requests. Experiments on a different simulation run to show the improvement over the state-of-the-art methods. The results indicate that DNA-MACP framework offers superior performance with all parameters.

## References

1. Pragaladan, R., Sathappan, S.: High confidential data storage using DNA structure for cloud environment. In: International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), IEEE Xplore. pp. 382–387 (2016). <https://doi.org/10.1109/CSITSS.2016.7779391>
2. Trombetta, A., Jiang, W., Bertino, E., Bossi, L.: Privacy-preserving updates to anonymous and confidential databases. *IEEE Trans. Dependable Secure Comput.* **8**(4), 578–587 (2011)
3. Dinga, J.-H., Chienb, R., Hungb, S.-H., Lina, Y.-L., Kuoa, C.-Y., Hsuec, C.-H., Chunga, Y.-C.: A framework of cloud-based virtual phones for secure-intelligent information management. *Int. J. Inf. Manag.* (Elsevier) **34**(3), 329–335 (2014)
4. Pragaladan R, Sathappan, S. 2017 A secure confidential data storage using fast chaos-based DNA cryptography (FSB-DC) for cloud environment. (Unpublished Article)
5. Ranalkar, R.H., Phulpagar, B.D.: DNA-based cryptography in multi-cloud: security strategy and analysis. *Int. J. Emerging Trends Technol. Comput. Sci. (IJETTCS)* **3**(2), 189–192 (2014)
6. Siddaramappa, V.: Data security in dna sequence using random function and binary arithmetic operations. *Int. J. Sci. Res. Publ.* **2**(7), 1–3 (2012)
7. Pragaladan, R., Sathappan, S.: Multi aspect sparse time integrated cut-off authentication (STI-CA) for cloud data storage. *Indian J. Sci. Technol.* **10**(4), 1–11 (2017). <https://doi.org/10.17485/ijst/2017/v10i4/107893>

8. Wenjuan, X., Zhang, X., Hongxin, H., Ahn, G.-J., Seifert, J.-P.: Remote attestation with domain-based integrity model and policy analysis. *IEEE Trans. Dependable Secure. Comput.* **9**(3), 429–442 (2012)
9. Oestreicher, K.: A forensically robust method for acquisition of iCloud data. *Digital Forensics Res. Conf. Digital Inv.* Elsevier **11**(2), S106–S113 (2014)
10. Carbanar, B., Tripunitara, M.V.: Payments for outsourced computations. *IEEE Trans. Parallel Distrib. Syst.* **23**(2), 313–320 (2012)
11. Yang, J., Xiong, N., Vasilakos, A.V., Fang, Z., Park, D., Xianghua, X., Yoon, S., Xie, S., Yang, Y.: A fingerprint recognition scheme based on assembling invariant moments for cloud computing communications. *IEEE Syst. J.* **5**(4), 574–583 (2011)
12. Sakr, S., Liu, A., Batista, D.M., Alomari, M.: A survey of large scale data management approaches in cloud environments. *IEEE Commun. Surv. Tutor.* **13**(3), 311–316 (2011)
13. Tu, M., Li, P., Yen, I.-L., Thuraisingham, B., Khan, L.: Secure data objects replication in data grid. *IEEE Trans. Dependable Secur. Comput.* **7**(1), 50–64 (2010)
14. Hao, Z., Zhong, S., Nenghai, Yu.: A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. *IEEE Trans. Dependable Secure Comput.* **23**(9), 1432–1437 (2011)
15. Hashema, I.A.T., Yaqooba, I., Anuara, N.B., Mokhtara, S., Gania, A., Khanb, S.U.: The rise of “big data” on cloud computing: review and open research issues. *Sci. Direct Inf. Syst.* (Elsevier) **47**, 98–115 (2015)
16. Shiraz, M., Gani, A., Ahmad, R.W., Shah, S.A.A., Karim, A., Rahman, Z.A.: A lightweight distributed framework for computational offloading in mobile cloud computing. *PLoS One* **9**(8), 1–10 (2014)
17. Habiba, U., Masood, R., Shibli, M.A., Niazi, M.A.: Cloud identity management security issues and solutions: a taxonomy. *Complex Adapt. Syst. Model.* **2**(5), 1–37 (2014)
18. Kim H., Chung, H., Kang, J.: Zero-knowledge authentication for secure multi-cloud computing environments. *Adv. Comput. Sci. Ubiquitous Comput.* Springer, 255–261 (2015)
19. Mehmet Sabir Kiraz: A comprehensive meta-analysis of cryptographic security mechanisms for cloud computing, Springer. *J. Ambient Intell. Humanized Comput.* **7**(5), 731–760 (2016)
20. Namasudra, S., Roy, P.: A new secure authentication scheme for cloud computing environment. *Concurr. Comput. Pract. Exp.* 1–20 (2016)

# Enhanced Secure Sharing of PHRs in Cloud Using Attribute-Based Encryption and Signature with Keyword Search



M. Lilly Florence and Dhina Suresh

**Abstract** Personal health record (PHR) is an emerging trend to exchange and share a person's health information with the help of the third party cloud providers. There are many researches done using Attribute-Based Encryption (ABE) technique to share the information securely. A perfect signature guarantees unforgeability and privacy for the signer. In Attribute-Based Signature (ABS) with a set of attributes given by the authority, a signer can authenticate a message with a predicate. We provide a novel method "Enhanced Secure Sharing of PHRs in Cloud using Attribute-Based Encryption and Signature with Keyword Search" which ensures security, scalability, efficiency. In our proposed scheme, (i) the health records are encrypted using ABE (ii) it is authenticated using ABS scheme (iii) further we allow the user to search the encrypted data using the keywords.

**Keywords** Attribute-Based Encryption · Attribute-Based signature Authenticating · Keyword search · Security

## 1 Introduction

Cloud is where the user is provided services by cloud service provider (CSP) [11, 12] where the user surrenders all his sensitive information to the third party cloud service provider. There are many cloud providers such as Google, Amazon, Microsoft. The cloud has different services such as SaaS, PaaS, and IaaS and different cloud

---

M. Lilly Florence (✉)

Adhiyamaan College of Engineering, Hosur 635109, Tamil Nadu, India  
e-mail: lilly\_swamy@yahoo.co.in

D. Suresh

St. Joseph's College of Arts and Science for Women, Hosur 635126, Tamil Nadu, India  
e-mail: dhinadulcy@gmail.com

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_34](https://doi.org/10.1007/978-981-10-7200-0_34)

375

models with which the individual or an industry can be benefited. It is accepted that cloud has many advantages such as cost efficiency, storage, easy to access, quick deployment, and so on. The main advantage is virtualization which is the pillar of cloud computing because it allows greater flexibility and utilization of the cloud. The vital issue in the cloud is that of security. Security in cloud can be enhanced by implementing cryptography. The data stored in the Internet may be sensitive. For security purpose, the data can be stored in an encrypted form to maintain its privacy. As there is a vigorous growth in information technology, the sensitive medical information is transferred from written records into electronic medical records. Personal health record (PHR) is an emerging trend to exchange and share a person's health information with the help of the third party cloud providers. The difference between a PHR and an EHR (electronic health record) is that in PHRs individuals have the ability to manage their own PHRs. A PHR that a doctor or a health plan provides would fall under the laws that protect medical privacy and set standards for maintaining the security of your medical information. This would include both HIPAA and the Confidentiality of Medical Information Act (CMIA) [1, 7]. Online PHRs ensure patient's information is available in emergencies and even when they are traveling. PHRs also make sharing and communication easier and faster. There are many proposed PHRs [14, 20] schemes. When PHRs are outsourced, there occur privacy concerns as the data is exposed to the clouds providers and unauthorized parties. To maintain the security and privacy, the data must be encrypted before outsourcing.

## 2 Literature Review and Related Work

### 2.1 Basic Encryption Algorithms

To provide secure communication over the network, [27] encryption algorithm plays a vital role. The following are the basic types of encryption. Symmetric: In symmetric key encryption, only one key is used for both encryption and decryption. The key is kept secret. DES, 3DES, AES, and RC4 are the commonly used symmetric encryption techniques. The symmetric key is classified into block cipher symmetric key encryption: where the input is taken as a block of plaintext of fixed size—DES and AES. Stream cipher symmetric key encryption: where one bit is encrypted at a time and so it is time-consuming. RC4 is an example of stream cipher. Asymmetric: It is also called as public-key encryption. It uses two keys, a public-key known to everyone and a private or secret key known only to the recipient of the message. RSA is an example of asymmetric encryption. Homomorphic: It is implemented in various public-key cryptosystems [3]. To process the data stored in the server, homomorphic encryption is useful.

## 2.2 *How to Share a Secret*

Adi Shamir introduced a cryptographic system [25] in 1979 where any two users can communicate securely and can verify each others signature. Their public-key and private key need not be exchanged. In this concept, let us consider the secret has to be given access to  $n$  number of people. Access is granted only if  $k \leq n$ . Let the secret be  $D$ .  $D$  is divided into  $n$  pieces  $D_1, \dots, D_n$ . We can make  $D$  easy computable if we have the knowledge of any  $k$  or more  $D_i$  pieces else it remains incomputable. If  $k = n$ , then all the members are needed to reconstruct the secret. Such a scheme is called a  $(k, n)$  threshold scheme the value of  $n = (2k - 1)$ .

## 2.3 *Identity-Based Encryption from Pairing*

The Identity-Based Cryptosystems and signature schemes [5, 6, 26] are a cryptographic system introduced in 1984 where any two users can communicate securely and can verify each others signature. The major differences between an identity-based system and a traditional system are authenticating the key, distributing the key, and using the key. The Identity-Based Encryption changed the technique by allowing the receivers public-key to be an arbitrary string like the e-mail ID of the receiver. The major drawback of IBE is it requires a centralized server and also a secure channel when transmitting the secret key.

## 2.4 *Attribute-Based Encryption*

In this cryptosystem scheme, the ciphertext and the keys are labeled with a set of descriptive attributes and the key can decrypt only if there is a match between the attributes of the ciphertext and the keys of the user. The concept of Attribute-Based Encryption was first proposed by Sahai and Waters [24]. ABE has the capacity to address complex policies. The exact lists of users need not to be known in previous. The knowledge of the access policy is sufficient. Encryption is based on the access structure, and decryption is possible only if the set of attributes matches the attributes of the ciphertext. An aspect of ABE is collusion resistance. Two or more users cannot combine their attributes to decrypt the ciphertext. Only an individual can access the data if his attributes satisfy the attributes of the ciphertext. There are two main types of ABE that are discussed below.

- Key Policy Attribute-Based Encryption: In this type of ABE, the ciphertext is generated by a set of attributes and the access policy is encoded with the users secret key [13, 30].
- Ciphertext Policy ABE: In this type of ABE, the attributes are embedded in the users secret key and the access policies are embedded in the ciphertext [4, 29].



### **3 Frame Work of Our Secure Sharing of PHR's in Cloud Using Attribute-Based Encryption and Signature with Keyword Search**

#### ***3.1 Problem Definition:***

There are many proposed PHRs schemes [17, 22]. Let us consider a cloud environment hosting the PHR service. Usually in a PHR service, there are three types of entities (i) the data owners/users, (ii) trusted authorities, and (iii) the cloud server. The data owner may be a patient who creates and signs his/her PHR records and stores it in the cloud server. There may be users who are curious to read others PHR's. There must be a guaranteed unforgeability and security [23]. In this fast-moving world where search engines play a vital role, why can't the user search the encrypted file using the keywords where each keyword is encrypted and signed in a secure way? To the best of our knowledge, there are less works done on this area where each keyword is encrypted and signed also the frequency for the keywords are generated for fast retrieval of records.

#### ***3.2 Our Contribution:***

- We have tried to provide a secure patient centric PHR access and efficient key management system.
- Our framework relies under the Multi Authority Cipher Text Attribute-Based Encryption (MA-CPABE) [9, 10] scheme.
- We took the Attribute-Based Signature (ABS) [15, 21] scheme to assure the data user that the owner himself has endorsed the message.
- Our scheme is built over multi authority settings.
- The attributes can be chosen by the data owner.
- An user can open the file of an owner only if his credentials are satisfied.
- Each keyword in the file is encrypted and signed.
- The user can search [8, 16, 31] the file using the encrypted and signed keywords if his credentials satisfy the owners access policy.
- We calculate the frequency of the keywords so that the files containing the more frequent occurring keywords will be given highest priority and the particular file with that keyword will be displayed first at the time of decryption.
- In our model, we can guarantee security, authenticity, privacy, and moreover fast searching of files using the keyword search.

### 3.3 Preliminaries

**Bilinear Maps** The field of Pairing-Based Cryptography [2] has exploded over the past years. Let  $G$  and  $G_T$  be two multiplicative cyclic groups of prime order  $p$ . Let  $g$  be a generator of  $G$  and  $e$  be a bilinear map, then  $e : G * G \rightarrow G_T$ . Useful bilinear maps have the following properties:

Bilinearity:  $\forall u, v \in G$  and  $\forall a, b \in \mathbb{Z}_q^*$  we have  $e(u^a, v^b) = e(u, v)^{ab}$ .

Non-degeneracy: If everything maps to the identity, that's obviously not interesting, so we have  $e(g, g) \neq 1$ .

Computability:  $e$  is efficiently computable.

**Access Structures** The tree approach was first described in Goyal, Vipul. Let  $P_1, P_2, \dots, P_n$  be a set of parties. An access structure  $A$  is a collection of nonempty subsets of  $P_1, P_2, \dots, P_n$ . An access structure  $A \subseteq 2^{P_1, P_2, \dots, P_n}$  is said to be monotone if for any  $B, C \in 2^{P_1, P_2, \dots, P_n}$ , if  $B \in A$  and  $B \subseteq C$  then  $C \in A$ .

### 3.4 Algorithm of the Proposed System

**Setup** The algorithm sets a wide range of parameters and outputs the public and private key depending on the attributes. Let  $\mathbb{A}$  be a universe set of attributes which is used to create the access policies [18] where  $A \subseteq \mathbb{A}$  and  $A = \{a_1, a_2, \dots, a_m\}$ . Using a randomized algorithm, we generate hash keys  $AH_i$  where  $i \geq 1$  for each attribute. Therefore,  $\{ah_1, ah_2, \dots, ah_i\} \leftarrow AtthashkeyGen(A)$  where  $A = \{a_1, a_2, \dots, a_m\}$  and  $A \subseteq \mathbb{A}$ . Fix two prime order groups  $G$  and  $G_T$ , a generator  $g$  and a bilinear map  $e : G * G \rightarrow G_T$ . We generate the system public-key  $PK$  and master secret key  $MSK$  using another randomized algorithm.

**User Policy Generation** The data owner creates his/her users  $\mathbb{U}$  where  $\mathbb{U} = \{u_1, u_2, \dots, u_n\}$  and sets the policy for each user so that the user whose credentials satisfy alone can view the file. This takes input the hash keys of the attributes, and the users are created with a valid mail id as the secret key for the user is sent through the mail id.  $\{up_1, up_2, \dots, up_n\} \leftarrow UserPolGen(\{ah_1, ah_2, \dots, ah_i\})$ . For example, the policy for user  $U_1$  may be  $U_1 = ((ah_1 AND ah_3) OR ah_6)$ . The policy issue is truly dependent on the data owner.

**File Encryption and Signature** Let  $F = \{f_1, \dots, f_d\}$  be a set of files to be encrypted and uploaded. The files are encrypted, and unique hash codes  $FH = \{fh_1, fh_2, \dots, fh_j\}$  where  $j \geq 1$  and each file is created using the public-key and the access policy. This can be called as the file or global signature. It is also used to identify the file.  $\{fh_1, fh_2, \dots, fh_j\} \leftarrow FilehashkeyGen(F, PK, \{ah_i\}_{ah \in U_n})$ . Each file hash key is identified by an identifier  $IDFH_j$ .

**File Process** There are two processes which are done on the file before it is encrypted. The common words are removed, and the remaining words are called keywords. These keywords are further converted to lower case. In the second step, we calculate the frequency of each keyword in a file. When the user searches the file with a keyword, the file with the highest frequency of the keyword will be displayed first.

- As we concentrate also on keyword search [19, 28], the file undergoes few processes. There is a predefined file  $SF$  where the commonly used words are listed. The file to be encrypted is compared with  $SF$ , and the common words are removed. It identifies the keywords  $KW = \{kw_1, \dots, kw_l\}$  and converts the words to lower case.  $\{kw_1, \dots, kw_l\} \leftarrow CaseConv(file, SF)$ .
- Just like our normal search engine, we would like to display the files according to the priority. The files are displayed by calculating the frequency of the keywords. The frequency of each keyword in a file is calculated using the frequency generation algorithm. Therefore,  $(kwfq_1, \dots, kwfq_r) \leftarrow KWfreqGen(kw_1, \dots, kw_l)$ .

**Keyword Encryption and Signature** As mentioned above, the file is encrypted with access policy. Each keyword in the file is encrypted with all the attributes in the access policy of the user, and therefore, signature for each keyword is created.  $\forall kw \in KW_l, \sigma CT_{kw} \leftarrow (kw, \{ah\}_{ah \in U_n})$ . Each keyword signature is identified by an identifier  $IDKW_l$ .

**Build Index and User Key Issue** The keyword index is built with the  $\sigma CT_{KW_l}$  for each hashfile  $fh_1, fh_2, \dots, fh_j$ . Therefore, the index contains  $BuildIndex \leftarrow (\sigma CT_{KW_l}, kwfq_r)$ .

The secret key is issued to the user  $USK \leftarrow KeyIssue(\{ah\}_{ah \in U_i}, MSK)$ .

**Decryption and Signature Verification** The user enters his secret key. First, authentication is done, and if the key is valid and the user credential is satisfied, then the user can enter any relevant keyword present in the file. The signature is verified and if the signature matches it searches over the index and the file is decrypted and displayed according to the frequency of the keyword.

$DecryptFile \leftarrow Searchkeyword(BuildIndex \leftarrow (\sigma CT_{KW_l}, kwfq_r), USK, ah_{ah \in U_i})$ .

## 4 A Small Challenge Game

### 4.1 Challenge

- Let a challenger run the *Setup* algorithm which gives the public-key and master secret key  $Setup \leftarrow (PK, MSK)$ . The algorithm selects  $(KW, IDFH_j, F)$  also the access policy  $\{UP\}_n$ . The challenger runs  $(IDFH_j, IDFH_j, F.KW)$  and gives  $(\sigma CT_{KW_l}, kwfq_r)$ .

- Process: The challenger gives  $USK$  according to the key generation. The  $USK$  contains  $MSK, \{UP\}_n$ . During the verification, the challenger returns  $Y \leftarrow verify(USK, \sigma CT_{KW_i}, ah_{ah_e U_i})$ .
- Challenge: Let the challenger have the keyword  $kw^*$  and  $up^*$  as the policy. The algorithm generates the key so that  $F(Keyissue, up^*) = 1$  will automatically generate  $USK^*$ . We say that the challenge is face and won if  $\leftarrow verify(USK, \sigma CT_{KW_i}, ah_{ah_e U_i}) = 1$  else it is considered that the game is lost.
- Theorem 3: Due to the one way hash function, our framework achieves keyword secrecy.
- Theorem 4: As security is guaranteed in ABE, our framework guarantees security, secrecy, and unforgeability.

To check with the feasibility of our system, we tried to upload files of different sizes. We calculated the upload time of the file, the time taken to search the keyword and decrypt the file. We have mentioned the file size and time in milli seconds and explained it in the graph Fig. 1. The notations used in the algorithms are mentioned in the Table 1.

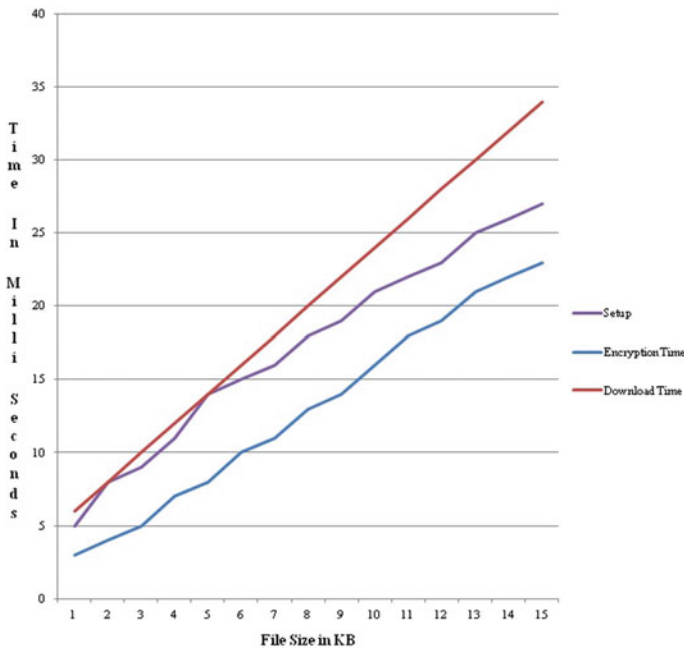


Fig. 1 Performance analysis of our framework

**Table 1** Notations

Frequently used notations	
$\mathbb{A}$	Universe set of attributes
$A = \{a_1, a_2, \dots, a_m\}$	Set of attributes
$AH = \{ah_1, ah_2, \dots, ah_i\}$	Attribute hash keys
$PK$	Public-key
$MSK$	Master secret key
$\mathbb{U} = \{u_1, u_2, \dots, u_n\}$	Set of users
$\mathbb{UP} = \{up_1, up_2, \dots, up_n\}$	Set of user policy olicity
$F = \{f_1, \dots, f_d\}$	Set of files
$FH = \{fh_1, fh_2, \dots, fh_j\}$	Set of hash files
$KW = \{kw_1, \dots, kw_l\}$	Set of keywords
$k_{wfq}$	Keyword frequency
$IDFH$	Hash file identifier
$IDKW$	Keyword signature identifier

## 5 Conclusion

We have introduced a novel framework called Secure Sharing of PHR's in cloud using Attribute-Based Encryption and signature with keyword search for secure cloud computing over encrypted data. This paper gives an overview of ABE and its evolution. In this paper, the evolution of Attribute-Based Encryption from the Identity-Based Cryptosystems and signature scheme is discussed.

## 6 Authors' Contributions

Both the authors contributed equally and extensively to the work presented in this paper.

**Acknowledgements** I would like to thank THE LORD MY SAVIOR for guiding and showering HIS blessings throughout my life. I take immense pleasure in thanking my guide Dr. M. Lilly Florence. I would like to thank my husband, parents, and my son for their patience and care.

## References

1. 104th United States congress, health insurance portability and accountability act of 1996 (HIPPA) (1996). <http://aspe.hhs.gov/admsimp/pl104191.htm>
2. Adida, B.: Special topics in cryptography. Instructors: Canetti, R., Rivest, R. Lecture 25: pairing-based cryptography (2004)

3. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Proceedings of crypto 98, pp. 26–45 (1998)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE symposium on security and privacy, pp. 321–334 (2007)
5. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Advances in cryptology Eurocrypt, vol. 3027, pp. 223–238. Springer, LNCS (2004)
6. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Proceedings of the 21st annual international cryptology conference on advances In: cryptology, pp. 213–229. Springer (2001)
7. California, Confidentiality of Medical Information Act (CMIA). [www.leginfo.ca.gov/cgi-bin/displaycode?section=civ-group=00001-01000](http://www.leginfo.ca.gov/cgi-bin/displaycode?section=civ-group=00001-01000)
8. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multikeyword ranked search over encrypted cloud data. In: Proceedings of IEEE INFOCOM (2011)
9. Chase, M., Chow, S.S.: Improving privacy and security in multi-authority attribute-based encryption. In: CCS 09, pp. 121–130 (2009)
10. Chase, M.: Multi-authority attribute-based encryption. In: The 4th theory of cryptography conference (TCC 2007) (2007)
11. Chen, Y., Paxson, V., Katz, R.H.: Whats new about cloud computing security?. Technical report UCB/EECS-2010-5. University of California at Berkeley, Electrical Engineering and Computer Sciences (2010)
12. Cloud computing security. <http://en.wikipedia.org/wiki/Cloud-computing-security>
13. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM conference on computer and communications security, pp. 89–98 (2006)
14. Ibraimi, L., Asim, M., Petkovic, M.: Secure management of personal health records by applying attribute-based encryption In: Technical report, University of Twente (2009)
15. Khader, D.: Attribute based group signatures. In: Proceedings of cryptology ePrint archive, Report 2007/159 (2007). <http://eprint.iacr.org/2007/159>
16. Li, M., Yu, S., Cao, N., Lou, W.: Authorized private keyword search over encrypted personal health records in cloud computing. In: Proceedings of ICDCS 11, Jun (2011)
17. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distr. Syst. **24**(1) (2013)
18. Liang, X., Lu, R., Lin, X., Shen, X.S.: Self-controllable access policy on phi in ehealthcare systems. In: Proceedings of AHIC 2010 (2010)
19. Liu, Q., Wang, G., Wu, J.: Secure and privacy preserving keyword searching for cloud storage services. J. Netw. Comput. Appl. **35**(3), 927–933 (2012)
20. Lohr, H., Sadeghi, A.R., Winandy, M.: Securing the e-health cloud. In: Proceedings of the 1st ACM international health informatics symposium ser IHI 10, pp. 220–229 (2010)
21. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures, In: Proceedings of CT-RSA11, vol. 6558, pp. 376–392. LNCS (2011)
22. Qian, H., Li, J., Zhang, Y., Han, J.: Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. IIIS, Springer (2014)
23. Racko, C., Simon, D.: Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Proceedings of crypto 91, pp. 433–444 (1991)
24. Sahai, A., Waters, B.: Fuzzy identity based encryption. In: Advances in cryptology—Eurocrypt, vol. 3494, pp. 457–473. Springer, LNCS (2005)
25. Shamir, A.: How to share a secret, 3rd edn, Commun. ACM **22**(11), 612–613 (1979)
26. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Proceedings of CRYPTO 84 on advances in cryptology, pp. 47–53, Springer, NY (1985)
27. Stallings, W.: Cryptography and Network Security: Principles and Practices, 4th edn (2006)
28. Sun, W., Yu, S., Lou, W., Hou, Y.T.: Your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans. (2016)

29. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Proceedings of cryptology ePrint 2008/290 (2011)
30. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proceedings of IEEE INFOCOM10 (2010)
31. Zheng, Q., Xu, S., Ateniese, G.: VABKS: verifiable attribute-based keyword search over out-sourced encrypted data. In: Proceedings of IEEE INFOCOM 2014 IEEE—ieeexplore.ieee.org (2014)

### Author Biographies



**Dr. M. Lilly Florence** Professor, Department of M.C.A, Adhiyamaan College of Engineering, Hosur. She received her Bachelor degree in Mathematics during 1995 and Master degree in Computer Application during 1998 at Manonmaniam Sundaranar University. She completed her M.Tech at Punjab University at 2003. She completed her Ph. D in computer science at Mother Teresa University during 2006–2011. She has published over 14 research papers in International and National journals.



**Dhina Suresh** was born in Tirunelveli, Tamil Nadu (TN), India, in the year 1983. She is working as an Assistant Professor, Department of Computer Science, St. Joseph's College of Arts and Science for Women, Hosur, Tamil Nadu (TN), India. She received the Master in Science (M.Sc) in Software Engineering degree from Periyar University, Salem, TN, India, in the year 2005 and Master of Philosophy (M.Phil.) of Computer Science Degree from the Periyar University, in 2007. Currently she is pursuing my Ph. D in Computer Science in Periyar University, Salem, Tamil Nadu (TN), India under the guidance of Dr. M.Lilly Florence. Her research area is security in cloud computing.

# Grey Wolf Optimization-Based Big Data Analytics for Dengue Outbreak Prediction



R. Lakshmi Devi and L. S. Jayashree

**Abstract** In recent decades, dengue fever (DF) and dengue hemorrhagic fever (DHF) outbreaks have occurred frequently in many tropical and subtropical regions of Asia. Big data-driven analytics are recently facilitated on the large dataset to monitor climate-driven changes and also for the dengue outbreak prediction. Many past studies have established an association between the meteorological variables and the dengue incidence. Hence, this paper examines the effects of meteorological factors on dengue incidence in one of the climatic categories of the tropical region in Asia. The nine meteorological parameters and number of dengue cases per week were considered in this study. Subsequently, the most influencing variables of dengue incidence were selected using a new heuristic optimization algorithm such as grey wolf optimization (GWO) based on Adaptive Neuro-Fuzzy Inference System (ANFIS) followed by negative binomial regression model which was employed to evaluate various lag times between dengue incidences and meteorological variables. Therefore, the derived meteorological variables with a time lag period are utilized for the big data analytics of dengue outbreak prediction.

**Keywords** Dengue • Feature selection • Optimization algorithm  
Grey wolf optimization • Negative binomial regression

---

R. Lakshmi Devi (✉)

Department of Computer Applications, S.A Engineering College,  
Chennai, India

e-mail: lakshmiddevir@saec.ac.in

L. S. Jayashree

Department of Computer Science and Engineering, PSG College  
of Technology, Coimbatore, India

e-mail: jayashreecls@gmail.com

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,

Advances in Intelligent Systems and Computing 645,

[https://doi.org/10.1007/978-981-10-7200-0\\_35](https://doi.org/10.1007/978-981-10-7200-0_35)



## 1 Introduction

One of the most dangerous worldwide emerging arboviral diseases is the DF [1]. The outbreaks mainly occur in many parts of tropical and subtropical regions. Various studies have stated that the endemic area of dengue extends over 60 countries [1–3]. World Health Organization (WHO) has stated that dengue fever causes more than 20,000 deaths per year and approximately 2.5 billion people live in dengue-endemic countries [4].

Previous studies have demonstrated statistically that there was a significant correlation between infectious diseases and meteorological factors such as rainfall, humidity, temperature, and wind speed [5–10]. Only limited statistical studies were available to describe the most influencing meteorological factors with a time lag period for the occurrence of dengue outbreak [11–13].

In soft computing, many feature subset selection techniques were available to select the most optimal features. It decreases the number of features resulting in speeding up of the computation/execution time. Therefore, in this study, new heuristic optimization algorithm, namely GWO, was applied to select the influencing meteorological factors for dengue outbreak.

The study will be most successful when data have been accumulated over long periods. To develop an early warning system, daily outcome data were desirable, but it was extremely difficult to gather meteorological and health data from the regions of developing countries [14]. Thus, the study motivated to use available online data to examine the most influencing meteorological factors on the dengue outbreak in the tropical region, and also, the analysis was carried out with weekly data.

In this paper, a distinguished climatic categorized region, such as tropical region, was used in the analysis of meteorological factors and dengue outbreak. The dataset consists of meteorological factors, namely average temperature, maximum temperature, minimum temperature, atmospheric pressure at sea level, average relative humidity, average rainfall, average visibility, average wind speed, maximum speed of the wind, and dengue outbreak in Colombo, Sri Lanka. In the past decade, there has been a dramatic increase of dengue incidences in Sri Lanka, mostly in urban and semi-urban regions. During the year 2012, there have been 40144 dengue cases and 131 deaths [15].

The main objectives of the proposed study are (i) to explore the most influencing meteorological factors using a new optimization algorithm GWO with ANFIS and (ii) to find the time lag period of each meteorological factor for the occurrence of dengue outbreak in the current week using negative binomial regression.

## 2 Grey Wolf Optimization

The GWO is initially proposed by Mirjalili et al. [16], and its algorithm is inspired by the democratic behavior and the hunting mechanism of grey wolves in the wild. In a pack, the grey wolves follow very firm social leadership hierarchy. The alpha

( $\alpha$ ) wolves are described as the leaders of the pack with male and female, which is considered as the fittest solution. The second level of grey wolves, which are subordinate wolves that help the leaders, is called beta ( $\beta$ ) which is known as the second best solution. Deltas ( $\delta$ ) are the third level of grey wolves which has to submit to alphas and betas, but dominate the omega, and this level of wolves is taken as a third best solution, respectively. The lowest rank of the grey wolves is omega ( $\omega$ ), which have to surrender to all the other governing wolves. The candidate solutions which are left over are taken as omega ( $\omega$ ). In the GWO, the optimization (hunting) is guided by alpha, beta, and delta. The omega wolves have to follow these,  $\beta$  and  $\delta$  wolves.

The grey wolves encircle prey during the hunt. The encircling behavior can be mathematically modeled as in Eqs. (1) through (4).

$$\vec{D} = \left| \vec{C} \cdot \vec{Y}_p(t) - \vec{Y}(t) \right| \tag{1}$$

$$\vec{Y}(t+1) = \vec{Y}_p(t) - \vec{A} \cdot \vec{D} \tag{2}$$

where  $t$  indicates the current iteration,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors,  $\vec{Y}_p$  is the position vector of the prey, and  $\vec{Y}$  indicates the position vector of grey wolves. The  $\vec{A}$  and  $\vec{C}$  vectors are calculated as in Eqs. (3) and (4)

$$\vec{A} = 2 \cdot \vec{a} \cdot r_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot r_2 \tag{4}$$

where components of  $\vec{a}$  are linearly decreased from 2 to 0 over the course of iterations as given in Eq. (5), and  $r_1, r_2$  are random vectors in [0, 1].

$$\vec{a} = 2 - t \cdot \frac{2}{\text{Max}_{iter}} \tag{5}$$

where  $t$  is the iteration number, and  $\text{Max}_{iter}$  is the total number of iterations allowed for the optimization.

The hunt is usually guided by the alpha, beta, and delta wolves, which have wide knowledge about the potential location of prey. The other search agents must update their positions according to best search agent's position. Equations (6) through (8) were used to update the search agents' positions.

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{Y}_\alpha - \vec{Y} \right|, \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{Y}_\beta - \vec{Y} \right|, \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{Y}_\delta - \vec{Y} \right| \tag{6}$$

$$\vec{Y}_1 = \vec{Y}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{Y}_2 = \vec{Y}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{Y}_3 = \vec{Y}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (7)$$

$$\vec{Y}_{(t+1)} = \frac{\vec{Y}_1 + \vec{Y}_2 + \vec{Y}_3}{3} \quad (8)$$

### 3 The Proposed Work

In this section, we present the proposed GWO optimizer based on ANFIS for feature selection. This study aims at assessing the impact of nine meteorological parameters on the prediction of dengue incidences and outbreaks in the tropical region. The data are collected from online source, Epidemiology Unit, Ministry of Health, Sri Lanka (<http://www.epid.gov.lk/>). The weekly epidemiological dengue reports of Colombo, Sri Lanka, were obtained for the period 2010–2014, and the meteorological data of the corresponding period are also available online and taken from <http://en.tutiempo.net/climate/sri-lanka.html> source. 260 weekly data were taken for the experiment: The training set consisted of 182 data points, and testing set consisted of 78 data points.

After the meteorological variables and dengue incidences, data are collected, and preprocessing was done on the datasets 1 and 2. Normalization is used to scale all the data in the same range of values [0, 1] for each input parameter. In this work, all input values in the dataset were normalized using Min-Max normalization as given in Eq. (9). Without normalization, training the real dataset will take adequate time for processing.

$$\text{Norm}(X_i) = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (9)$$

where  $X_{\min}$  is the minimum value of input parameter X, and  $X_{\max}$  is the maximum value of input parameter X.

#### 3.1 Development of GWO for Feature Selection on Dengue Dataset

Feature selection is an approach used in machine learning for choosing a subset of relevant features for achieving robust prediction systems that provide a better understanding of data by offering their significant characteristic features. In designing prediction systems, feature selection plays a vital role in selecting the relevant enhanced prediction accuracy [17].

**Table 1** Parameter setting for the feature selection

Parameter	Value
No. of search agents (wolves)	100
No. of iterations	100
Problem dimension	9
Search Domain	The given dataset

The new heuristic algorithm GWO was applied for extracting the most influencing meteorological features on dengue. In Table 1, optimizer-specific parameter setting is outlined. All the parameters are set according to domain-specific knowledge.

The initial wolves population for an N-dimensional optimization problem was characterized by 1 X N array which is defined as in Eq. (10)

$$\text{wolves} = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9] \tag{10}$$

where  $f_i$  is the meteorological features used in the datasets 1 and 2. N is the number of features in the dataset. The initial values for the features were done randomly. Each of the original features was treated as a wolves position in the search space. The search for the optimal feature subset was an update of each wolves position in the search space.

During the training process, each wolves position represents one feature subset. The selected feature subset from the dengue dataset was fed as input to the basic ANFIS predictor. For each candidate subset, the prediction error rate was calculated using fitness function. In this paper, a prediction error rate minimization-based fitness function was used and updated at each iteration. The fitness function is given in Eq. (11).

$$\text{Fitness} = \sqrt{\frac{1}{N} \sum_{i=1}^N |Y_{\text{actual}} - Y_{\text{predicted}}|^2} \tag{11}$$

where N is the number of data points. At the end of each iteration, the first three best solutions were considered as positions of alpha, beta, and delta wolves and the remaining omega wolves positions were updated based on the positions of alpha, beta, and delta wolves, respectively. The process continued till the maximum number of iterations reaches. Finally, the minimized error rate was considered as an optimal meteorological feature subset.

The convergence speed for GWO is ensured for its efficient searching capability and for the simplicity of the used fitness function. GWO algorithm results are compared with particle swarm optimization (PSO) and genetic algorithm (GA) as they are known for their popularity in space searching.

### 3.2 Finding Time Lag Period Using Negative Binomial Regression Model

After selecting the most influencing meteorological variables using GWO algorithm, the work was extended to find the time lag period for the derived meteorological variables. Poisson and negative binomial regression models appear to be appropriate when the output variable (number of dengue cases) consists of non-negative integer [15]. In our case, the number of dengue cases (per week) rarely exhibit equal means and variances. Therefore, negative binomial regression was used as the model to find the time lag period by using SAS Version 12.0 for Windows (SAS Institute Inc., Cary, North Carolina, USA).

The estimation of the parameters was done by maximum-likelihood method to choose the best model. The final model was selected based on Akaike information criterion (AIC), Bayesian information criterion, (BIC), and log-likelihood values. The model with least AIC value was selected as the best model.

## 4 Results and Discussion

The performance of the proposed GWO algorithm was compared with other meta-heuristics algorithms such as GA and PSO with nine input variables. ANFIS was used to evaluate the fitness of different feature subsets. The superiority of selected optimization techniques was proven by running the datasets for 100 iterations each. By using GWO, GA, and PSO, the selected features are shown in Table 2.

The evaluation results of fitness function by GA, PSO, and GWO with ANFIS predictor are given in Table 3. The fitness values indicated that GWO achieves better performance than other optimization algorithms, which ensures the searching

**Table 2** Selected features using different optimizers

Optimizer	No. of features	No. of selected features	Selected features
GA	9	3	Average temperature, average wind speed, maximum speed of the wind
PSO	9	4	Minimum temperature, average wind speed, RF, maximum speed of the wind
GWO	9	5	Average temperature, minimum temperature, average rainfall, average relative humidity, average wind speed

**Table 3** Evaluation of fitness function by applying different optimizers

	Training dataset			Testing dataset		
	GA	PSO	GWO	GA	PSO	GWO
Mean fitness	0.23	0.25	<b>0.23</b>	0.24	0.24	<b>0.24</b>
Std. deviation fitness	0.02	0.03	<b>0.01</b>	0.03	0.02	<b>0.01</b>
Best fitness	0.23	0.24	<b>0.23</b>	0.23	0.24	<b>0.22</b>
Worst fitness	0.25	0.26	<b>0.25</b>	0.26	0.26	<b>0.25</b>

capability of the GWO. From Table 3, it is revealed that GWO optimizers have lowest mean fitness as well as have lowest standard deviation of the obtained fitness values that establish the optimizer stability and convergence.

The finding results show that the most effective dengue influencing features can be used in developing an early warning system for dengue surveillance and prevention and also to facilitate the public health officials to gain knowledge about the influencing factors of dengue virus transmission and not getting confused by many features which were not necessarily effective. The output from the proposed GWO with ANFIS predictor shows that the feature selection by GWO has a good impact on the accuracy of ANFIS.

Subsequently, the appropriate time lag period of selected variables was evaluated by using negative binomial regression, which explores the most predominant time periods for the occurrence of the current dengue cases. Six models were constructed with different combinations of time lag periods from zero to three weeks on each selected parameter. The AIC, BIC, and log-likelihood values of six models were compared, and the model with lowest measuring terms was considered as a best-fit model.

Based on the p-values, measuring terms were identified that the average temperature, rainfall and wind speed of two weeks lag period for dengue incidence at 1%, 1% and 5% level respectively. The same result was also inclined by Wan et al. [15]. Therefore, these were the most emphasized periods for the occurrence of dengue outbreak in Colombo. Other parameters, minimum temperature and humidity, were not considered as they have negative correlation with an output variable (number of dengue cases).

## 5 Conclusion and Future Direction

Prediction error rate minimization-based objective function was used in GWO to find optimal meteorological features which cause dengue in tropical regions. The proposed GWO algorithm was used to find the optimal positions in the complex

search space through the interaction of individuals in the population. The performance of the proposed algorithm was compared with GA and PSO over a set of dengue case data repository, and the proposed approach proves better performance in terms of minimizing the error rate.

The features selected by GWO were average temperature, minimum temperature, average relative humidity, average rainfall, and average wind speed which have an optimal position with least fitness value, and the two weeks of lag period were the most significant period for the occurrence of dengue outbreak that was evaluated by using the negative binomial regression. The selected meteorological factors are further used for dengue outbreak prediction which yields 96.65% of prediction accuracy. In the future work, socioeconomic factors, mosquitoes breeding data, census data, etc., can be included to identify the significant association between these factors and dengue outbreak.

## References

1. Rajapakse, S., Rodrigo, C., Rajapakse, A.: Treatment of dengue fever. *Infect Drug Resist.* **5**, 103–112 (2012)
2. Rasgon, J.L.: Dengue fever: mosquitoes attacked from within. *Nature* **476**, 407–408 (2011)
3. Brady, O.J., Gething, P.W., Bhatt, S., Messina, J.P., Brownstein, J.S., et al.: Refining the global spatial limits of dengue virus transmission by evidence-based consensus. *PLoS Negl Trop Dis* **6**, e1760 (2012)
4. WHO: Dengue guidelines for diagnosis, treatment, prevention and control: World Health Organization. 1–147 p (2009)
5. Egbendewe-Mondzozo, A., Musumba, M., McCarl, B.A., Wu, X.: Climate change and vector-borne diseases: an economic impact analysis of malaria in Africa. *Int. J. Environ. Res. Public Health* **8**, 913–930 (2011)
6. Huang, F., Zhou, S., Zhang, S., Wang, H., Tang, L.: Temporal correlation analysis between malaria and meteorological factors in Motuo County, Tibet. *Malar J* **10**, 54 (2011)
7. Haque, U., Hashizume, M., Glass, G.E., Dewan, A.M., Overgaard, H.J., et al.: The role of climate variability in the spread of malaria in Bangladeshi highlands. *PLoS ONE* **5**, e14341 (2010)
8. Traerup, S.L., Ortiz, R.A., Markandya, A.: The costs of climate change: a study of cholera in Tanzania. *Int. J. Environ. Res. Public Health* **8**, 4386–4405 (2011)
9. Xu, L., Liu, Q., Stige, L.C., Ben Ari, T., Fang, X., et al.: Nonlinear effect of climate on plague during the third pandemic in China. *Proc. Natl. Acad. Sci. USA* **108**, 10214–10219 (2011)
10. Ari, T.B., Gershunov, A., Tristan, R., Cazelles, B., Gage, K., et al.: Interannual variability of human plague occurrence in the Western United States explained by tropical and North Pacific Ocean climate variability. *Am. J. Trop. Med. Hyg.* **83**, 624–632 (2010)
11. Fairos, W.Y.W., Azaki, W.H.W., Alias, M., Wah, Y.B.: Modelling dengue fever (DF) and dengue hemorrhagic fever (DHF) outbreak using Poisson and negative binomial model. *World Acad. Sci. Eng. Technol. Int. J. Math. Comput. Nat. Phys. Eng.* **4**(2) (2010)
12. Ahmed, S.A., Siddiqi, J.S., Quaiser, S., Kamal, S.: Using PCA, Poisson and negative binomial model to study the climatic factor and dengue fever outbreak in Lahore. *J. Basic Appl. Sci.* **11**, 8–16 (2015)

13. Siriyasatien, P., Phumee, A., Ongruk, P., Jampachaisri, K., Kesorn, K.: Analysis of significant factors for dengue fever incidence prediction. *BMC Bioinform.* **17**, 166 (2016)
14. Honda, Y., Ono, M.: Issues in health risk assessment of current and future heat extremes. *Glob. Health Action* 2 (2009)
15. Epidemiology Unit, Ministry of Health, Sri Lanka. <http://www.epid.gov.lk/web/index.php>
16. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimization. *Adv. Eng. Softw.* **69**, 46–61 (2014)
17. Kira, K., Rendell, L.: The feature selection problem: traditional methods and a new algorithm. In: *Proceedings of AAAI'92*, San Jose, CA, AAAI (press) (1992)



# Design of Smart Traffic Signal System Using Internet of Things and Genetic Algorithm



P. Kuppusamy, P. Kamarajapandian, M. S. Sabari and J. Nithya

**Abstract** The revolution of Internet of Things facilitates innumerable dimensionalities over industrial, home, and business uses. The amalgamation of sensors and manhandling devices with existing infrastructure makes the process proficient and reduces the manpower processing time. The novel smart traffic signal system is proposed using smart server with cloud-oriented infrastructure to improve the signal processing time in road intersection traffic signal post that reduces the waiting time, jamming, and contamination. This proposed approach also observes the vehicle mobility to change the signal for providing the way and also used to track vehicles. This tracking process would be utilized to find vehicles that are involved in illegal transportation and accident due to high speed. Genetic algorithm is proposed to observe multi-location data and analyze single-point well-designed decision using vehicles queue at four-direction roadway signal intersection. The experiments have been demonstrated with Arduino Uno kit and evaluated the smart traffic signal system by comparing with normal traffic system. The results show that proposed system facilitates hassle-free travel by decreasing the waiting time for the green signal and accidents.

**Keywords** Internet of Things · Traffic · Sensors · Cloud · Roadway Vehicles

## 1 Introduction

Rapid progress of Information and Communications Technology (ICT) is performing major role in people's lifestyle. Many people are utilizing the smart objects like smartphone, laptop that gather, process, share, analyze, and store the data with

---

P. Kuppusamy (✉)

Madanapalle Institute of Technology and Science, Madanapalle, India  
e-mail: drpkscse@gmail.com

P. Kamarajapandian · M. S. Sabari · J. Nithya  
Gnanamani College of Technology, Namakkal, India

© Springer Nature Singapore Pte Ltd. 2018

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,  
Advances in Intelligent Systems and Computing 645,  
[https://doi.org/10.1007/978-981-10-7200-0\\_36](https://doi.org/10.1007/978-981-10-7200-0_36)

395

the aid of ICT. Aberer et al. [1] have mentioned that the human living style is transformed apparently since the mobile device communication developments with business, industry, and social network, i.e., Twitter and Facebook. Internet of Things (IoT) is an inspiring technology that is developed based on ICT. It provides vast opportunity for the researchers through recent Internet technology since that covers Internet Protocol (IP)-based embedded objects such as laptops, smartphones armed with sensors (e.g., mobility sensors, camera, light-dependent resistor, and pressure sensor), Global Positioning System (GPS), radio-frequency identification (RFID) readers, environment sensing, machines, and building automation devices. Atzori et al. [3] have presented the method in terms of invention and progress of IoT, and it is noteworthy in modern scenario and size of the objects is also compact. These diminutive devices are inspiring the IoT evolution to connect, communicate, operate, and control objects around users with its distinctive IP address. Users who act as object handlers in IoT and physical objects (e.g., documents, vehicles, parts, devices, etc.) focused, positioned, selected various functionalities. Things/objects play as communicators and also accumulate the data about its environment to observe and map phenomena of common interest. These objects act as a gateway to communicate with the environment. The objects share the cache content to the neighbor objects to improve the data sharing and replace the stale cache content in regular interval [15]. The IoT intends to yield more facilities to the users through sensing the environment factors and send data, provide comprehensive report with visualization tools.

## 2 Related Work

Hernandez-Munoz et al. [10] have stated the features of IoT that includes small storage, less processing speed, reliability, privacy, and security. Gartner presented that IoT will contain 30 billion smart objects interconnected to observe, communicate, operate, and potentially trigger the interconnected devices by 2020. Hui et al. have been offered [11] 6Low-power Wireless Personal Area Network (6LowPAN) protocol that enables devices to connect, share, and operate smart objects such as laptop, sensors, smart manageable camera, and lightbulb. Debasis and Jaydip [7] described services that are applied on different fields such as video surveillance, traffic monitoring, logistics, air traffic, environmental, home [13]. Vermesan et al. [19] proposed IoT European Research Cluster (IERC) scheme that is road map for forthcoming research and development beyond 2020. Research and Innovative Technology Administration [18] was proposed intelligent transportation approach to provide swift decisions using IoT-based systems for designing smart traffic-less environment. Intelligent Transportation System (ITS) and IntelliDrive strategic research approaches are integrated with ICT and road transportation convergence vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies [14].

Choosri et al. [6] have proposed traffic light control design to improve traffic lights signal system changing time that reduced the vehicle congestion using RFID technology. Promila [17] proposed an intelligent traffic light and density control

that reduced the traffic by considering traffic density in the track. Jara et al. [12] presented security scheme to avert the traffic monitoring and control framework from numerous attacks in urban cities. Agent-based fuzzy logic technology was proposed to predict the hefty traffic flow using vehicle movements [2, 4]. The genetic algorithm (GA) is an optimization method to compute best candidate fitness solution over the collection of perceived candidate solutions [16]. Gubbi et al. [9] described that combination of smart objects with industry applications increased the requirement of distributed computing processing power. The researchers and government organizations take effort on ITS strategic design that inspires us to work on smart traffic control design by collecting vehicular mobility information, process, analyze, and share based on the mobilization of roadway [18].

### 3 Smart Traffic Signal System with IoT

The smart traffic signal is proposed by mixing IoT with current roadway infrastructure to diminish travel time, contaminant emission, traffic, and easing driving stress. Traffic signals are assembled at cross-point of the roadway in every town and city. The motorbike, car, bus, auto-rickshaw, logistics carriers are mostly exploited for traveling and goods shipment. This leads to hefty jamming in the roadway.

Figure 1 shows the proposed IoT-based smart traffic control system that contains local smart server (LSS), cloud server (CS), sensors that are located in the traffic signal and tracks of the road and cloud, respectively. The sensors are fixed to observe real-time vehicle mobility in the road track at each direction of intersection in traffic signal. The sensors are placed in each track to sense 200 m from the signal lamp. The traffic signal is designed with IoT and sensor-based devices such as RFID tag, RFID sticker readers, parking area sensors, and camera systems. The traveler contains smartphone, laptop, and RFID tag-based number plates fixed on vehicles. The tracks are alienated into cells in that sensor nodes are fixed around a traffic signal intersection. Sensor objects observe vehicle's position either slight movement or waiting position on the road at the traffic signal. The observed location is transferred to a local smart server (LSS) via their neighbor nodes. The LSS comprises powerful processor, more internal memory, and storage than sensors. Hence, the received data are processed and transmitted to the LCD display that is located 2 km in front of the traffic signal. In the proposed system, LSS-based sensor system provides the real-time context-aware computing and event stream processing using Wi-Fi (Wireless Fidelity) connectivity. IoT objects are transferring the data to constrained application protocol (CoAP) [5]. CS and Web server are connected using transmission control protocol (TCP), Hyper Text Transfer Protocol (HTTP), and IP using high-speed fiber optical communication for facilitating high bandwidth data transmission. The LSS is sending the analyzed traffic data to the CS that is stored for remote access and effective guidelines. The roadway travelers also exploit the CS through Web to update traffic status.

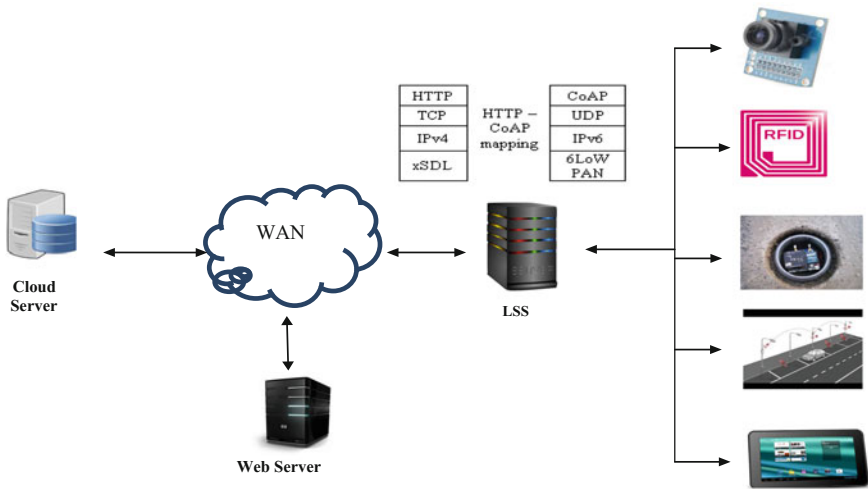


Fig. 1 Smart traffic control system with IoT

The proposed system is also used genetic algorithm (GA) for optimization that selects the high fitness value from the gathered data. The vehicles are moving in leftmost, rightmost tracks that permit to take turn only on left and right route directions, respectively. The RFID reader parses the register number plate, and camera contains the photograph of vehicles that information sends to the LSS that shared with CS to track the vehicles along the traffic signals in the route. The green light glows in one route direction even unavailing of vehicle movement in the track, though more vehicles are awaited often for the green signal in other route direction. The LSS gathers the data from sensors of four-sided route tracks and computes the vehicle density using fitness function. Hence, it changes the onset of green, red lights based on the vehicle density that reduces the jamming in more vehicle mobility direction. The route direction of traffic signal is mentioned as  $d$ ,  $t$  denotes the tracks in each route direction of the traffic signal,  $W_T$ ,  $W$  is meant for number of vehicle (density) in the track, and total number of vehicles in all directions,  $l, r$ , represent the left and right turning of vehicle movement, respectively.  $S$  represents the sensors fixed on the tracks in all route directions of the roadway. The green signal track and red signal directions are specified as  $i, j$ , respectively. Vehicles are waiting to take left and right turn that is calculated using Eq. (1). A total number of vehicles waiting in all directions are computed as given in Eq. (2), and maximum vehicle density among all directions is determined using Eq. (3).

$$W_T = W_l + W_r \tag{1}$$

$$W = \sum_{i=1}^n d_i W_T + \sum_{j=1}^n d_j W_T \tag{2}$$

$$f = \max(d_i W_T, d_j W_T), i \neq j \text{ and } i = 1, j = 2 \dots n \quad (3)$$

### Algorithm

RFID reader scans the vehicle registration number

Camera records the vehicles movement in LSS

Sensors senses the vehicles movement in the left, right tracks ( $l, r$ )

$d_i W_l, d_i W_r$  represents data collected from sensor  $S_i$  fixed in the tracks  $l, r$  in route  $d_i$

Sensors collect the data of the vehicle movement in  $d_i, d_j$  at regular intervals  $t$ .

Compute the maximum vehicles mobility  $f$  from all tracks of four route directions

$d_i W_T$  represents density of the vehicles in route  $d_i$

if  $d_i W_T$  value in  $i > d_j W_T$  in  $j$  then

green light is glow to this route  $d_i$  track

vehicles heading towards the destination

red light glow in  $d_j$

else

red light is glow in  $d_i$

vehicles wait for the green light

LSS stores the data in CS for analysis and web access

This proposed system exploits the optimization data processing method that increases the data accessibility effectively by choosing the individual parameter of sensing data. It saves the atmosphere by diminishing the carbon monoxide emission since it diminishes waiting time and traveling delay.

## 4 Results and Discussion

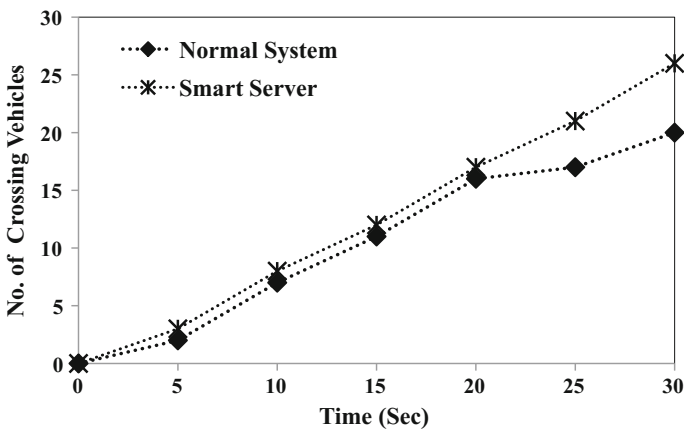
The experimental setup is established for proposed smart traffic system that contains Intel Core i3 2 GHz processor, 4 GB RAM HP notebook, Arduino Uno controller [8] that connects IoT smart devices such as RFID tag, sticker, light-emitting diode

(LED), passive infrared (PIR), RTC timer, seven segment display, and ultrasound sensors. Wiznet Ethernet shield (Mac address 20:47:47:04:64:93) for LAN, XAMPP open source software is used to demonstrate the smart system that contains software like Apache Web server, PHP, and MySQL for Web page and database design. This model has been studied, and implementation is demonstrated for analyzing real-time traffic by processing the data of traffic signal at four-road intersection in Salem, Tamil Nadu, India, on September 2016. This existing infrastructure is compared with smart server-based traffic model with respect to traffic at normal and peak hours. The number of vehicles are passed through intersection in a day with respect to normal and peak hours have been listed in Table 1. The signal is delayed to switch over lamp (red to orange, orange to green, green to red) based on the number of vehicle movements in the direction of roadway. In present infrastructure, signal is also changed in cyclic method continuously without considering the vehicles availability, i.e., the presence of vehicle movement or absence of vehicles. Hence, smart server-based traffic framework system is taken the parameters such as number of vehicles crossing the signal in each track and signal changing frequency (number of times signal changed) with respect to time.

Figure 2 shows an average number of vehicles crossing in a day for 30 s in single track of the direction. The smart server system is permitted vehicles like

**Table 1** Four directions vehicles mobility at intersection

Direction (d)	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>
Normal hours (10.00 AM–04.00 PM)	839	956	701	824
Peak hours (08.00 AM–10.00 AM) and (04.00 PM–08.00 PM)	1023	1147	987	1012



**Fig. 2** Number of crossing vehicles versus time

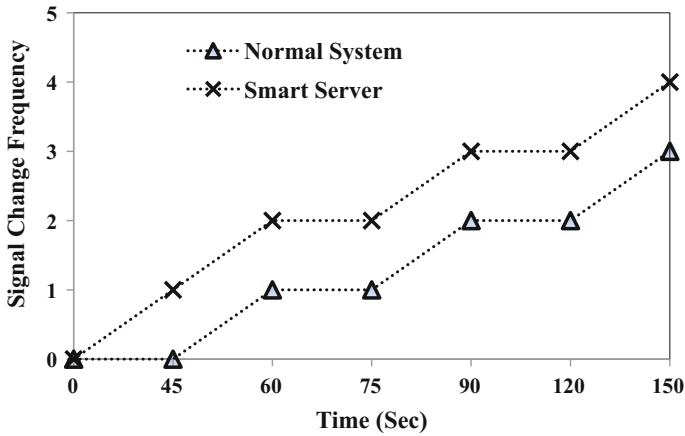


Fig. 3 Traffic signal changing frequency versus time

normal system with slight improvement till 20 s and more vehicles from 21 to 30 s. The result is shown that more number of vehicles crossed the signal in smart server-based system than normal system since tracks are established with sensor and optimized data processing at LSS.

Figure 3 illustrates the signal changing frequency with respect to time interval. It is considered 15, 3, 5, 7 vehicles in  $d_1, d_2, d_3,$  and  $d_4,$  respectively. The result is taken by varying time from 0 to 150 s. The proposed system computes the vehicle density using sensors. Hence, the red and green signals are onset frequently in different direction tracks. In existing system, signal is changed one time per 60 s even though more vehicles are in the  $d_1$  track. But proposed model is changed, and the signal at 45 s for the same track since fitness is computed based on vehicle

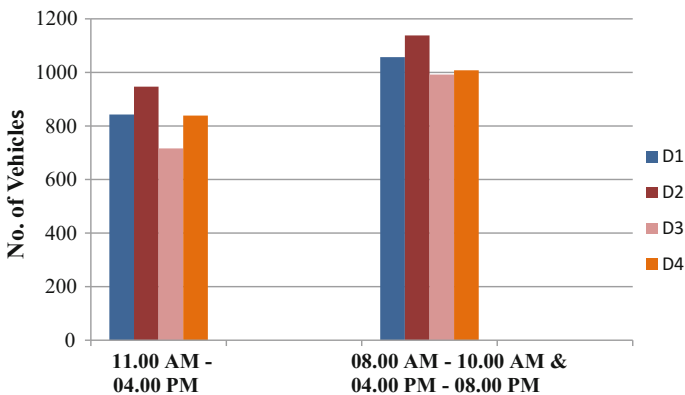


Fig. 4 Number of passing vehicles at peak and normal hours

density. The result shows that the proposed system changes the signal at 45 s itself since more vehicles at  $d_1$  than other tracks

Figure 4 illustrates the total number of vehicles that passed the signal in each direction  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$ , respectively, at normal hours (11.00 AM–04.00 PM) and peak hours (08.00 AM–10.00 AM and 04.00 PM–08.00 PM). The result shows that more number of vehicles are passed that include car, bus, lorry, motor bike, auto-rickshaw in direction  $d_3$  than other directions since it is toward industries.

## 5 Conclusion

This research work highlights the importance and necessity of smart traffic control system for smoothening the traffic and diminishes the delay in the traffic signal. The proposed system is integrated with existing infrastructure using IoT devices for smart process. This framework is considered several factors and requirements of traffic system like vehicle density, events, rainfall, and civic amenities. The smart server has processed the data using optimized method genetic algorithm that controls the signal based on density efficiently. The cloud server is used to store the data and smooth the remote access at anywhere, anytime. The travelers update the traffic status using smartphone through online that avoids the jamming and delay. The experimental results show that proposed smart system reduces the jamming, waiting time, traveling delay and increases the vehicle mobility than normal system. This work will be extended in forthcoming research work using big data analytics to analyze and security for cooperative traffic environment in the smart city.

## References

1. Aberer, K., Hauswirth, M., Salehi, A.: Infrastructure for data processing in large scale interconnected sensor networks. In: Proceedings of the International Conference on Mobile Data Management, Germany, pp. 198–205 (2007)
2. Al-Sakran, H.O.: Intelligent traffic information system based on integration of internet of things and agent technology. *Int. J. Adv. Comput. Sci. Appl.* **6**(2), 37–43 (2015)
3. Atzori, L., Lera, A., Morabito, G.: The internet of things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
4. Bhadra, S., Kundu, A., Guha, S.K.: An agent based efficient traffic framework using fuzzy. In: Proceedings of the International Conference on Advanced Computing & Communication Technologies (2014)
5. Castellani, A.P., Loreto, S., Rahman, A., Fossati, T., Dijk, E.: Best practices for HTTP-CoAP mapping implementation. <https://tools.ietf.org/html/draft-constellani-core-http-mapping-07> (2013)
6. Choosri, N., Park, Y., Grudpan, S., Chuarjedton, P., Ongvisesphaiboon, A.: IoT-RFID testbed for supporting traffic light control. *Int. J. Inf. Electron. Eng.* **5**(2) (2015)
7. Debasis, B., Jaydip, S.: Internet of Things—applications and challenges in technology and standardization. *Wireless Pers. Commun.* **58**, 49–69 (2011)
8. Doukas, C.: Building Internet of Things with the Arduino, vol. 1 (2012)



9. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. *Elsevier Future Gener. Comput. Syst.* **29**(7), 1645–1660 (2013)
10. Hernandez-Munoz, J.M., Vercher, J.B., Munoz, L., Galache, J.A., Presser, M., et al.: Smart cities at the forefront of the future internet. In: *The Future Internet*, pp. 447–462. Springer, Berlin. <http://dl.acm.org/citation.cfm?id=1983741.1983773> (2011)
11. Hui, J., Culler, D., Chakrabarti, S.: 6LoWPAN: incorporating IEEE 802.15.4 into IP architecture—internet protocol for smart objects (IPSO) alliance, White Paper #3. <http://www.ispo-alliance.org> (2009)
12. Jara, A.J., Varakliotis, S., Skarmeta, A.F., Kirstein, P.: Extending the internet of things to the future internet through ipv6 support. *Mob. Inf. Syst.* **10**(1), 3–17 (2014). IOS Press
13. Kuppusamy, P.: Smart home automation using sensors and internet of things. *Asian J. Res. Soc. Sci. Humanit.* **6**(8), 2642–2649 (2016)
14. Kuppusamy, P., Kalaavathi, B.: Novel authentication based framework for smart transportation using IoT and memetic algorithm. *Asian J. Res. Soc. Sci. Humanit.* **6**(10), 674–690 (2016)
15. Kuppusamy, P., Kalaavathi, B., Chandra, S.: Optimal data replacement technique for cooperative caching in MANET. *ICTACT J. Commun. Technol.* **5**(3), 995–999 (2014)
16. Manpreet, B., Jyoti, S., Ravneet, K.: Energy efficient data gathering in wireless sensor network using genetic algorithm. *Int. J. Recent Innovation Trends Comput. Commun.* **3**(7), 4830–4833 (2015)
17. Promila, S.: Intelligent traffic light and density control using IR sensors and microcontroller. *Int. J. Adv. Technol. Eng. Res.* **2**(2) (2012)
18. Research and Innovative Technology Administration, Policy White Paper: Achieving the vision: from VII to intellidrive, ITS JPO U.S. DOT. [http://www.its.dot.gov/research\\_docs/pdf/2From%20VII%20to%20IntelliDrive.pdf](http://www.its.dot.gov/research_docs/pdf/2From%20VII%20to%20IntelliDrive.pdf) (2010)
19. Vermesan, O., Friess, P., Guillemin, P., et al.: Internet of things strategic research roadmap. The Cluster of European Research Projects, Tech. Rep. <http://www.internet-of-thingsresearch.eu/pdf/IoT> Cluster Strategic Research Agenda 2009.pdf (2009)

# An Innovated SIRS Model for Information Spreading



Albin Shaji, R. V. Belfin and E. Grace Mary Kanaga

**Abstract** Epidemic models do a great job in spreading information over a network, among which the SIR model stands out due to its practical applicability, with three different compartments. When considering the real-world scenarios, these three compartments have a great deal of application in spreading information over a network. Even though SIR is a realistic model, it has its own limitations. For example, the maximum reach of this model is limited. A solution to this is to introduce the SIRS model where the nodes in the recovered (removed) state will gradually slip into the susceptible state, based on the immunity loss, which is a constant. This presents the problem because in the real-world scenario, this immunity loss rate is a dependent parameter so a constant won't do justice. So to cope with the real-world problem, this paper presents a variable called immunity coefficient, which is dependent on the state of the neighbors.

**Keywords** Epidemic models · Multilayer network · Information cascade

## 1 Introduction

Social networks have become a mean to pass on the information these days, and people depend on the digital world more than ever to figure out what is going on in the real world. This has given a unique opportunity for governments and organizations to spread their propaganda among a large audience. This spreading process in computer science is called diffusion. Diffusion in a network is the process of

---

A. Shaji (✉) · R. V. Belfin · E. Grace Mary Kanaga  
Department of Computer Science and Engineering, Karunya University,  
Coimbatore, India  
e-mail: albinshaji@karunya.edu.in

R. V. Belfin  
e-mail: belfin@karunya.edu

E. Grace Mary Kanaga  
e-mail: grace@karunya.edu

spreading information, or virus, or infection in a network starting from a single seed node or multiple seed nodes. The process of diffusion is very important in sociology and epidemiology. By studying diffusion over a social network, one can identify how fast a particular message is being reached to a community and what is the maximum reach of information, etc. For decades, many scholars have studied the process of information diffusion to devise a more efficient diffusion model. The first step toward the study of diffusion was always discovering the association that existed between the participants. The participants along with the relation existed between them formed a network.

As the complexity of the network increased, diffusion has become hard to achieve. Studying the diffusion of a real-world multilayered network is a herculean task, so Javier Borge-Holthoefer et al. [1] proposed an alternative modeling approach to understand the dynamics of diffusion. This paper tries to imitate the real-world scenario, by introducing a variable value that depends on the neighbors to determine whether the candidate node will become recovered or susceptible. Implementation of this model is done after studying the general SIR and existing versions along with the SIRS model. This helped us to understand the benefits of existing systems as well as how they differ from a real system. This work intends to reduce the gap between SIR diffusion models and real diffusion process.

## 2 Related Works

Over decades, studies have been done to determine the methods to improve information diffusion over multilayered networks. In standard SIR model, each node in one of the three states or compartments, susceptible, infected, or recovered (removed), hence the term compartmental model. If a participant is in the susceptible (S) state, it indicates that the node has not undergone any infection [2–4]. A node in the infected (I) state can spread infection or information or rumor to its neighbors in susceptible state at a rate  $\beta$ , the infection rate. Any node in the infected state recovers after a time  $\tau$ . If the participant is in the recovered (R) state, it can no longer spread the infection. Reference [2] used a fixed population to come up with the following equations:

$$\frac{dS}{dt} = -\frac{\beta SI}{N} \quad (1)$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma \quad (3)$$

where

$$N = S(t) + I(t) + R(t) \tag{4}$$

shows how clustering will affect information propagation, by forming clusters of nodes within networks, and has figured that as the probability of a node joining a cluster increases ( $\alpha$ ) final epidemic size also increases [5]. That is if the network is smaller and densely connected chance for triggering, information epidemic becomes greater when compared to a huge network, which is loosely connected. The standard SIR model is based on the analogous properties of networks and nodes, but real social contact networks exhibit heterogeneous qualities. The novel ISIR deterministic epidemic model using the nonlinear force of infection is considering the crowding or protection effect to address the limitation of standard SIR model where the linear fixed force of infection used [6]. In improved SIR (ISIR), the infection rate is not a fixed value; instead, it is a function of the number of infected individuals.

$$\lambda = \beta(I) \tag{5}$$

Thus, the equation for ISIR became:

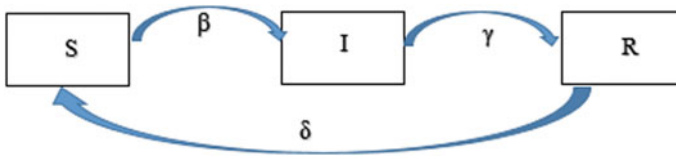
$$\frac{dS}{dt} = -\lambda(I)S \tag{6}$$

$$\frac{dI}{dt} = \lambda(I)S - \gamma I \tag{7}$$

$$\frac{dR}{dt} = \gamma I \tag{8}$$

Even in this model, the average degree of nodes has a great deal of influence in diffusion. As average degree increases, epidemic spreading is more efficient. One of the important factors in SIR is the recovery rate or how fast an infected node gets disinfected. The ‘hazard rate,’  $h(t)$ , is the rate of recovery at time  $t$  conditional on surviving until time  $t$ , and as proposed in [7, 8], it will be a monotonically decreasing function. The fractional order model is biologically motivated by the observation that in some disease processes, the longer a person is infectious, the more likely they remain infectious.

A control mechanism for SIR model with a state-dependent switching cost has recently been introduced [9]. The authors introduced a minimum threshold value  $y_m \geq 0$ , over which a control action is needed. The idea is to obtain an optimal control, which is applicable everywhere satisfying the condition  $0 \leq u(t) \leq U$ , and final time  $T > 0$  that minimizes the cost index. This optimal control mechanism is found to have an intermediate control action over cases when the epidemic spread is



**Fig. 1** Single node SIRS model

low, thus maintaining a reduced economic cost. A Markov chain modeling of SIRS epidemics on complex networks is shown in [10].

Figure 1 shows three states and corresponding transitions where  $\beta$  is the transmission probability,  $\gamma$  is the healing probability, and  $\delta$  is the immunization loss probability. Reference [11] proposed a new epidemic SIRS model with the discrete delay on the scale-free network, with a basic reproduction number  $R_0$  obtained during mathematical analysis. If  $R_0 < 1$ , then the disease-free equilibrium is globally attractive and if  $R_0 > 1$ , then the disease-free equilibrium is unstable, and the disease is uniformly persistent. This study revealed that the spreading speed of infection in cooperative systems could be found using the corresponding linearized system. The spatial modeling of epidemics is yet to be done. Reference [12] is a further study of the traveling wave phenomenon and devised a time-periodic traveling wave model for diffusive SIR epidemic model.

They summarized the existence of a time-periodic traveling wave in the SIR model. Taking the crowding or protection effect into account, an ISIR model was devised [9], which characterized the epidemic spread on social contact network, by employing stochastic and q-influence models to simulate epidemic propagation. This ISIR epidemic model helps to predict the epidemic dynamics and optimize the control strategies by evaluating their performance in simulations.

### 3 Proposed Methodology

The SIR epidemic model infects the nodes with information, which starts randomly and proceeds to its neighbors. But, when the time moves the infection tends to wear off from the nodes and the nodes will shift to cured or recovered state. So after a particular time may be minutes, or hours, or days, the information will fade out of all the nodes in the network. Consider a case where an actor wants to promote a product in his workplace, and he picks some influencers among his work community. Think a situation where there is no ‘word of mouth’ factor happens, and all of the influencers completely forget the message. This situation will be a nightmare for the actor who attempted to promote his product, isn’t it? So it is indeed important that the information or idea, which one tried to propagate, should impale

to more nodes for a certain amount of time, so that many people will have the information. The continuous propagation is not possible in SIR model. In SIR model, the information will be completely lost in a certain amount of time. This paper presents a model where information remains with at least some nodes, and chances of adoption to an innovation are present.

### 3.1 Innovated SIRS Model for Closed Systems

The model proposed in [5] has the same compartments as S the susceptible, I the infected, and R the recovered. Any node in S is susceptible to infection with infection rate  $\beta$ ; nodes in the state I will recover at a rate  $\gamma$ ; and the recovered node will have an immunity value. The concept here is that this immunity will wear off in time, and the node will become susceptible to infection again. The rate at which immunity wears off can either be a constant provided or could be calculated from the state of neighbors. The mathematical formulation of this model is as follows:

$$dS = -\beta S + \delta R \tag{9}$$

$$dI = \beta S - \gamma I \tag{10}$$

$$dR = \gamma I - \delta R \tag{11}$$

where  $\delta$  is the immunity loss rate and takes a constant value. Figure 1 is a graphical representation of this model. The issue with taking a standardized immunity loss rate is that it does not do justice to the real-world scenario. In real-world examples, each node is under constant influence from its neighbors, so ignoring the state of one node’s neighbors, while dealing with the state change of a node is not sound. Similarly, there is a force that drives the state change of every node, the infection rate. Therefore, it is safe to say that the rate at which a node in recovered state changes into the susceptible state is dependent on the infection rate and the number of infected neighbors of that node. The model proposed in this paper has a variable and at any given time  $t$ , for a node ‘i,’

$$\delta = |N_i \cap I(t)| * \frac{\beta}{|N_i|} \tag{12}$$

where  $\beta$  is the set of neighbors of node ‘i,’  $I(t)$  is the set of all infected nodes at time  $t$ , so  $|N_i \cap I(t)|$  is the number of infected neighbors of node ‘i’ at time  $t$ . The  $\delta$  value changes for every node at every time.

**Table 1** Simulation setup R programming

Parameter	Value
Initial population	1001
Infection rate	1.4247
Recovery rate	0.14286

### 4 Implementation Details

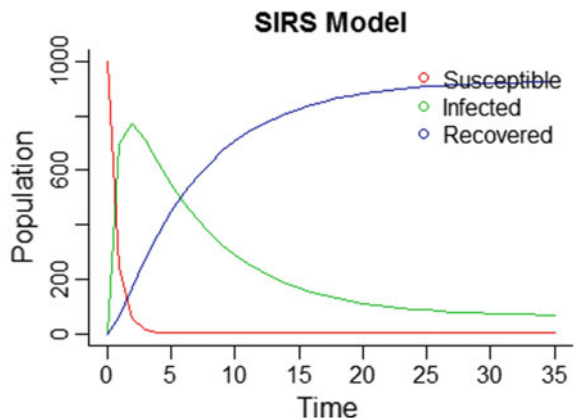
By randomly selecting the nodes for infection, the simulation shows the spreading process at every tick. Each individual in the setup has 5% chance of starting infected. If a node is infected at a tick, at the next tick, any susceptible person nearby can get infected. For every person whose state changes from infected to cured, immunity is set. Over time, it is reduced by the immunity-loss parameter. When the immunity value of a node becomes zero, it changes its state from cured to susceptible.

The epidemic model simulated in R programming used a 1001 node network, among the 1001 nodes, 1000 belong to the susceptible state, and 1 is the infected state. Initially, no nodes are in the recovered state. Table 1 is the initial simulation setup of the closed system.

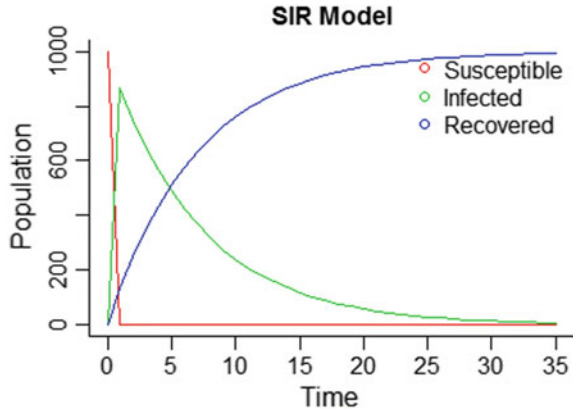
### 5 Numeric Results and Graphs

Figure 2 is a graph that plots the population of nodes in susceptible (red), infected (green), and recovered (blue) states against time in the SIRS model. From this graph, it is clearly visible that after a certain time, the infected and recovered nodes reach a saturation point. Thus, leaving some nodes with infection. This is advantageous when doing promotion works. Figure 3 is a graph of nodes in each compartment of SIR model when plotted using the same parameters from Table 1. While comparing these two graphs, one can clearly see that in SIR model, all the nodes are converted into the recovered state. But our innovated SIRS model has

**Fig. 2** Graph of the infected and not infected population over time



**Fig. 3** Graph of the infected and not infected population over time



some nodes still sticking to the infected state. While comparing these two graphs, one can clearly see that in SIR model, all the nodes are converted into the recovered state.

But our innovated SIRS model has some nodes still clinging to the infected state. After running the simulation, the number of secondary infections that has arisen due to a single infected node, introduced in a wholly susceptible population is obtained as 4.26. Since the nodes to be infected next are selected at random, this value may change at every execution. Table 2 contains the numerical results obtained from the SIRS simulation.

**Table 2** Simulation setup R programming

Susceptible	Infected	Recovered
1000	1	0
240	697	63
58	772	170
15	711	170
5	628	366
3	550	446
3	481	515
3	422	574
4	371	625
4	327	668
4	290	705
5	258	737
5	230	765
5	169	825
5	141	853
6	130	864
6	113	881
6	100	894
6	91	903



Looking into the population distribution, it is clear that as the time passes, population reaches a constant state. Every compartment has some nodes in it, which continue the diffusion process. Even though the process seems to be over, the nodes continue to change their state maintaining a constant ratio. This shows that there are always some nodes with the information.

## 6 Conclusion

‘An idea is like a seed, once you plant it deep enough no matter what it will grow into a big tree.’ In a diffusion process, the success is determined not only by the reach of diffusion but also to check whether the idea was planted deep enough to grow big. The shortcoming of SIR epidemic model is that it isn’t planted deep enough. Introduction of the immunity parameter and the variable immunity loss rate ensures that the idea is planted deep enough to stick along in the innovated SIRS model. This model is suited best for promotion works because promotion is successful when the ‘word of mouth’ starts spreading and people start adapting to the new idea. This model helps users to plan the promotion strategies, simulate the reach, and get an expected outcome from each strategy. This model is helpful for social workers to strategize how to organize, awareness programs in an efficient manner. Since the simulations allow one to test different scenario, choosing the best among them won’t be any difficult.

## References

1. Borge-holthoefer, J., Baños, RA., González-bailón, S.: Cascading behaviour in complex socio-technical networks. 3–24 (2013). <https://doi.org/10.1093/comnet/cnt006>
2. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. In: Proceedings of the royal society A: mathematical, physical and engineering sciences (1927). <https://doi.org/10.1098/rspa.1927.0118>
3. Li, T., Liu, Q., Li, B.: The analysis of an SIRS epidemic model with discrete delay on scale-free network. 1939–1946 (2015)
4. Liu, Q., Jiang, D.: The threshold of a stochastic delayed SIR epidemic model with vaccination. *Physica A* **461**, 140–147 (2016). <https://doi.org/10.1016/j.physa.2016.05.036>
5. Martinčić-Ipšić, S., Margan, D., Meštrović, A.: Multilayer network of language: a unified framework for structural analysis of linguistic subsystems. *Physica A* **457**, 117–128 (2016). <https://doi.org/10.1016/j.physa.2016.03.082>
6. Ruhi, N.A., Hassibi, B.: SIRS epidemics on complex networks: concurrence of exact markov chain and approximated models. *Cdc* (2015). [arXiv:1503.07576](https://arxiv.org/abs/1503.07576) <https://doi.org/10.1109/CDC.2015.7402660>
7. Wang, H., Wang, X.S.: Traveling wave phenomena in a kermack mckendrick SIR model. *J. Dyn. Diff. Equat.* **28**(1), 143–166 (2016). <https://doi.org/10.1007/s10884-015-9506-2>
8. Wang, Z.C., Zhang, L., Zhao, X.Q.: Time periodic traveling waves for a periodic and diffusive SIR epidemic model. *J. Dyn. Diff. Equat.* (2016). <https://doi.org/10.1007/s10884-016-9546-2>

9. Chai, W.K., Pavlou, G.: Path-based epidemic spreading in networks. *IEEE/ACM Trans. Networking* **25**(1), 565–578 (2017). <https://doi.org/10.1109/TNET.2016.2594382>
10. Zhang, Z., et al.: Modeling epidemics spreading on social contact networks. *IEEE Trans. Emerg Top. Comput.* **3**(3), 410–419 (2015). <https://doi.org/10.1109/TETC.2015.2398353>
11. Zhuang, Y., Yağan, O.: Information propagation in clustered multilayer networks. *IEEE Trans. Netw. Sci. Eng.* 1–14 (2015). <https://doi.org/10.1109/TNSE.2015.2425961>
12. Angstmann, C.N., Henry, B.I., McGann, A.V.: A fractional order recovery SIR model from a stochastic process. *Bull. Math. Biol.* **78**(3), 468–499 (2016). <https://doi.org/10.1007/s11538-016-0151-7>