A Project Report
on

PREDICTING STUDENT PERFORMANCE WITH DEEP NEURAL NETWORKS

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

Under The Supervision of
Dr.D Rajesh Kumar
Professor
Department of Computer Science and Engineering

Submitted By-(BT4056)

18SCSE1140024 - ANMOL CHHETRI

18SCSE1010256 - UTSAV SHARMA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the projec, entitled **"PREDICTING STUDENT PERFORMANCE WITH DEEP NEURAL NETWORKS."** in partial fulfillment of the requirements for the award of the Bachelor of Technology,submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY,2021** to **DECEMBER,2021**, under the supervision of Dr.D Rajesh Kumar, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Anmol Chhetri,18SCSE1140024
Utsav Sharma,18SCSE1010256

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name-Dr. D Rajesh Kumar
Designation-Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Anmol Chhetri:18SCSE1140024 and Utsav Sharma:18SCSE1010256 has been held on _____ and his/her work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:
Place:

Table of Contents

Abstract

In present educational systems, student performance prediction is getting worsen day by day. Predicting student performance in advance can help students and their teacher to keep track of progress of a student. Many institutes have adopted continuous evaluation system today. Such systems are beneficial to the students in improving performance of a student. The purpose of continuous evaluation system is to help regular students. In recent years, Neural Networks have seen widespread and successful implementations in a wide range of data mining applications, often surpassing other classifiers.

This study aims to investigate if Neural Networks are a fitting classifier to predict student performance from Learning Management System data in the context of Educational Data Mining.To assess the applicability of Neural Networks, we will compare their predictive performance against six other classifiers on this dataset. These classifiers are Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine and Logistic Regression and will be trained on data obtained during each course.

Python is a full featured for general purpose programming language. Python host numerous open source libraries and almost all general-purpose libraries for machine learning which can be further use for deep learning models. All this benefits from the python ecosystem lead to the top two libraries for numerical analysis of deep learning was developed for python language, that is Tensorflow and Theano library.
TensorFlow is an open source library for computing numerical using data flow graphs. The data flow graphs are also known as Static Computation graph. A developer must first design the input layer and connect every input layer to the hidden layer then the same from hidden layer to output layer.it will be useful to deep neural network model for effective prediction.

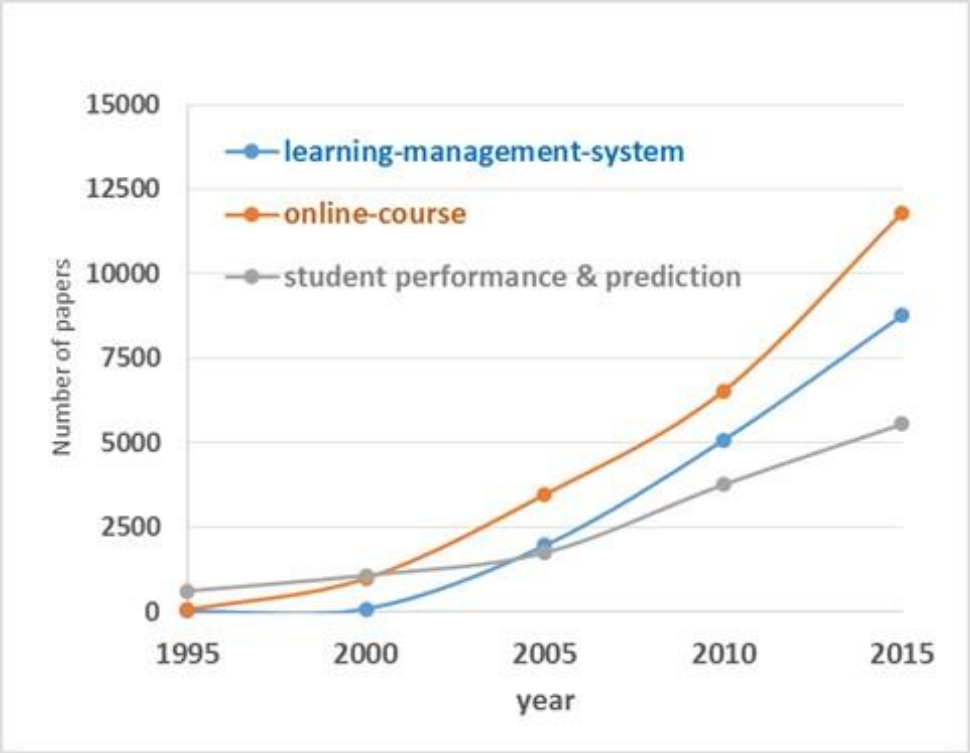
The goal of this study is to assess to what extent Neural Networks can be used to predict student performance: assessing whether they did or did not need academic assistance, based on LMS data.
We will show that predictive performance of the Neural Network on all courses at once exceeded that of other classifiers in terms of accuracy, while being on par with other classifiers interms of recall.

CHAPTER-1
INTRODUCTION

This section will introduce the background and problem statement of this study as well as the formulated research questions.

Background:

In recent years, the use of internet-based educational tools has grown rapidly as well as the research surrounding them (see below diagram).



Above figure denotes Number of papers in Educational Data Mining related fields. source: Google Scholar.

These tools provide a clear advantage for students and teachers alike, with the ability to access and share course data from anywhere in the world, track student progress and provide rich educational content. These tools generate vast amounts of data obtained in a non-obtrusive manner that can give a better look into the way students learn and interact with course materials. The challenge is to put these data to good use to improve on the educational process.One of the purposes these data

can be used for is the prediction of whether a student is going to pass or fail a course. Being able to predict student performance enables a teacher or educational institution to provide appropriate assistance to students that are at risk to miss the mark. Assisting them in a timely manner will reduce the number of students failing a course and may indirectly reduce the amount of students dropping out of their educational program.This is a societal interest that can have a positive impact on students, parents, teachers and
educational institutions alike.

The analysis and extraction of information and patterns from vast amount of data is called Data Mining. When the data comes from an educational setting we are dealing with a subdo?main of data mining called Educational Data Mining, or EDM. This is a field of research that applies data mining, statistics and machine learning to data derived from educational environ?ments. It seeks to extract meaningful information from vast amounts of raw data that can be used to improve and understand learning processes (Scheuer & McLaren, 2012). In order to extract interesting information, like predicting if a student requires academic assistance, we can make use of machine learning algorithms that can automatically predict this outcome based on
the data. In the field of EDM, a wide set of machine learning algorithms have already been used to various degrees of success like Naive Bayes Classifiers, k-Nearest Neighbors, Random Forests, Decision Tree Classifiers, Support Vector Machine algorithms and Neural Networks (Romero &Ventura, 2010).

The family of classifiers this study focuses on, Neural networks, have shown promising results in domains like speech recognition (Graves & Jaitly, 2014), computer vision (Venugopalan etal., 2014), recognizing music (Costa, Oliveira, & Silla, 2017), playing complex games like GO(Wang et al., 2016) and economic forecasting (Nametala, Pimenta, Pereira, & Carrano, 2016),but their use in EDM has thus far been limited compared to other classification algorithms (Baker & Inventado, 2014).This can be partly explained by their difficulty to set up, the lack of convenient all in one packages that are easy to use and the often long training times (Gaur,
2012) . But they do offer clear benefits over other machine learning algorithms. They are able to classify instances in domains that are not linearly-separable and can handle noisy and complex data (Schmidhuber, 2015). These properties make them especially suited for a domain like EDM where the data, given the fact that it is based on human behvaior, can be complex, might contain irrelevant entries as well as non linear relations. Assessing their applicability for the EDM domain by

comparing their performance to that of other classifiers could therefore result in new insights.

The aims to assess the accuracy and recall performance of Neural
Networks, while using all available predictors, compared to that of a majority baseline and 6
other classification algorithms:

- Naive Bayes
- k-Nearest Neighbors
- Random Forest
- Decision Tree
- Logistic Regression
- Support Vector Machine

CHAPTER-2
LITERATURE SURVEY

The following section wil start with an overview of the current state of EDM. This will be followed by an analysis of Neural Networks.

Educational Data Mining:
Research performed in this study can be classified under EDM. EDM is a sub-group of data mining that focusses on researching, developing and applying various automated methods to explore large-scale data coming from educational settings. This is done to increase the understanding of the way students learn, study educational questions and improve the effectiveness of teaching and learning activities. This goal is achieved by transforming the raw data into information that can have a direct impact on educational practice and research.

Neural Networks in Student Performance Prediction
Despite certain downsides, Neural Networks show promising results in EDM. They have been applied to student performance prediction, with various studies plotting them against other machine learning algorithms. However, differences are observed between studies in terms of performance, used predictors, sample size, data transformation, number of distinct courses and number of labels that need to be predicted.Sample size, by which we mean the number of students in a dataset, can influence the generalizability and portability of a model . Many EDM studies exist that make use of large samplesize to predict student performance, which use algorithms like Decision Trees, Bayesian Classifiers or Support Vector Machines. Good examples of which are a study by Jayaprakash et al. (2014) using a sample of 15150 students and another study with data from 10330 students (Kabakchieva, 2013). But studies involving Neural Networks have, as far as we have found, only used smaller sample sizes,with the maximum encountered being 649 students, in a study by Cortez et al (2008). Other studies we encountered relied on even smaller sample sizes for their predictions. Agrawal and Pandya(2015) used 100 students and Moucary et al.(2011) 73 students. Although the performance reported in these Neural Network studies are high, the portability of their findings is hard to determine. The network trained on 73 students might achieve good results for those
students, but it might fare significantly worse if presented with new students that

might come from another university or from another academic year. More students in a dataset should result in a more diverse sample and could therefore improve generalizability of the findings (Payne &Williams, 2005).

Classifiers:

In order to have a reference to compare the performance of the Neural Network against, we used six different classifiers to perform predictions on the data. These are k-Nearest Neighbors,Naive Bayes, Support Vector Machine, Logistic Regression, Decision Tree and Random Forests.These algorithms have all seen global use in EDM over the past years (Romero & Ventura,2010).

- The k-Nearest Neighbors (k-NN) algorithm looks at what known instances are close to theinstance we want to predict to perform classification. To perform optimally, the k parameter(number of neighbors) needs to be tuned. The larger the value of k gets, the less influence noise will have on the classification, but the decision boundaries between different classes will become less separable . In the context of student performance prediction, the algorithm obtained an accuracy of 57% on a dataset of 10330 students predicting five different performance labels (bad, average, good,very good and excellent) and an accuracy of 62.9% on a dataset of 566 students where it predicted a pass or fail .

- The Naive Bayes classifier is a simple probabilistic classification algorithm which main advantages are its speed, simplicity and versatility. It is popular as a baseline algorithm . One of the caveats of this algorithm is that it assumes that all predictor variables are independent, which might not always be the case.

- Support Vector Machines classify data by constructing a hyperplane in high-dimensional space that separates the classes . Using a special technique SVMs can also achieve non-linear classification. They have shown robust practical performance in a broad variety of domains like image recognition, text mining and bioinformatics. They have obtained accuracies of 86.3% on a sample of 395 students where five labels had to be predicted,and 86.26% on a sample of 15150 students where a pass or fail was predicted.

- Logistic regression is a regression model that has a categorical variable as output. Although not frequently mentioned in the EDM studies we encounterd, with one occurence in Stapel et al(2016) where it obtained an accuracy of 68.2%, it is widely used in other data mining domains and is therefore included in this study.

- Decision Tree classifiers are a predictive modelling approach that uses a tree like structure for its representation. Decision Trees are often used to identify the most optimal strategies to reach a certain target in real-world settings because their output is easily transformable in step by step directions (Baker & Inventado, 2014). This property, combined with their fast training time (Rokach & Maimon, 2014) explains their widespread use in EDM. Jayaprakash et al.(2014) obtained a prediction accuracy of 85.92% with the Decision Tree, on a dataset of 15150 students by predicting a pass or fail. Kabackhieva (2013) only managed an accuracy of 63.1% on a dataset of 10330 students predicting five levels of student performance. The difference in performance can be explained by the difference in setup and purpose of the two studies. The study by Kabakhieva (2013) seeks to devise a model that predicts academic performance, which serves as a tool to determine if someone should be admitted to a specific academic program. As such, it only uses historic data like grades and statistics obtained during the student's previous educational program, and more general statistics like age and gender. Additionally, it seeks to predict five different labels, ranging from bad to excellent performance, which can have an impact on the predictive accuracy. The study by Jayaprakash et al. (2014) seeks to devise a model that predicts whether or not a student will pass a course, and serves as an early alert system. They also use predictors from previous educational programs, and combine them with predictors gathered from the LMS course data. A variation on the Decision Tree classifier is the Random Forest which is an ensemble technique that uses a collection of Decision Trees to obtain a prediction.

The accuracy and number of classes that need to be predicted vary widely between studies.A recapitulation of results from studies involving binary classification, which have to predict two labels (for example: pass or fail, or requires assistance, or doesn't require assistance) can be found in Table 1. The variability in accuracies among studies could be attributed to the diversity and size of the data and the quality and type of the predictors that were used. Some studies use predictors that were gathered only during previous educational programs, like previously obtained grades (with different variations) or if a student has repeated a class. Other studies use these predictors suplemented with data gathered during a course, like time spent online or number of documents opened.

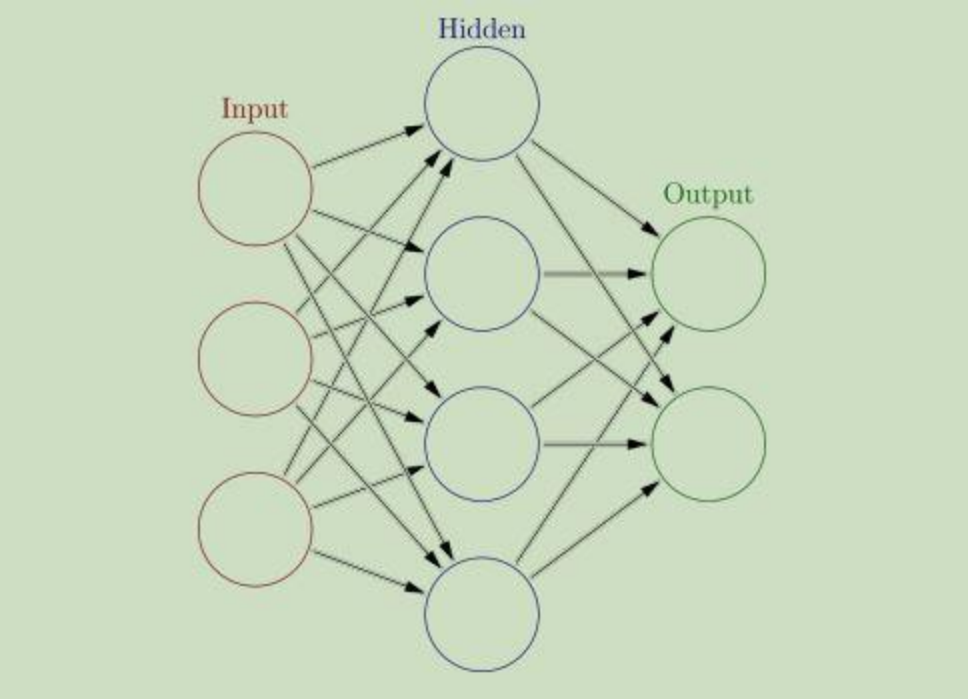
Table 1: Classification accuracies from studies predicting 2 labels (pass or fail)

	Sample	NN	SVM	DT	RF	NB	k-NN	LR
Jayaprakash, et al. (2014)	15150	-	86.3%	85.9%	-	84.1%	-	-
Stapel, et al. (2016)	566	-	-	71.5%	67.9%	65.4%	62.9%	68.2%
Agrawal, et al. (2015)	100	97.0%	-	91.0%	96.0%	80.0%	-	-
Calvo, et al. (2006)	240	80.2%	-	-	-	-	-	-

NN: Neural Network, SVM: Support Vector Machine, DT: Decision Tree, RF: Random Forest,NB: Naive Bayes, k-NN: k Nearest Neighbors, LR: Logistic Regression.

Neural Networks

The classifier this study focuses on to predict student performance belongs to the family of Neural Networks. Neural Networks are algorithms that mimic the way our brain works. They consist of an array of interconnected nodes that exchange information among each other, comparable to the way our neurons, connected by dendrites and axons, exchange information. They learn iteratively over time by observing different examples, similarly to how children can learn skills from their parents by observation. However, unlike children that can learn recognize and object after only observing it once, Neural Networks often require a greater set of observations to attain sufficient predictive capacity, as they are notoriously data hungry. Neural Networks differ from other classification algorithms in the fact that internally, information is processed in a parallel way, comparable to how our brain functions. This differs from the serial processing that many other algorithms like Decision Tree classifiers use.



The structure of a Neural Network .

The property that makes Neural Networks interesting for complex domains like computer vision, playing complex games like GO or understanding human speech, is their ability to derive answers from complex and imprecise data. Neural Networks are able to detect patterns that are too complicated for humans or other machine learning algorithms to pick up. And have been used successfully in numerous business applications for pattern recognition, prediction and classification. These properties make them particularly suited for a domain like EDM where large quantities of data are available and where the data is often noisy and not linearly separable owing to the fact that it is based on human behavior. Neural Networks also have certain disadvantages. It is at present impossible to derive how a network came to a certain output. Compared to a Decision Tree classifier in which one can follow a set of steps to arrive at a classification, a Neural Network doesnt store any explicit representation of the way it achieved its result. As such, a Neural Network is often referred to as a black box where information goes in, and a result comes out. They might therefore not be suited in some cases where the decision process needs to be explicit. This would be the case with expert system giving medical advice for which the decision process needs to be checkable. In our case, this black box nature is not a dealbreaker, as the consequence of mislabeling someone is not critical and will in most cases not require explicit explanation. Another drawback, is that Neural Networks require

vast amounts of data to achieve satisfactory performance and are therefore not suitable for every dataset . However, given the size of the dataset
used in this study, with records of 4601 students, this will most likely not be an obstacle.

Neural Networks in Student Performance Prediction

Despite certain downsides, Neural Networks show promising results in EDM. They have been applied to student performance prediction, with various studies plotting them against other machine learning algorithms. However, differences are observed between studies in terms of performance, used predictors, sample size, data transformation, number of distinct courses and number of labels that need to be predicted. Sample size, by which we mean the number of students in a dataset, can influence the gen?eralizability and portability of a model. Many EDM studies exist that make use of large samplesize to predict student performance, which use algorithms like Decision Trees, Bayesian Classifiers or Support Vector Machines. Good examples of which are a study by Jayaprakash et al. (2014) using a sample of 15150 students and another study with data from 10330 students (Kabakchieva, 2013). But studies involving Neural Networks have, as far as we have found, only used smaller sample sizes,with the maximum encountered being 649 students, in a study by Cortez et al (2008). Other studies we encountered relied on even smaller sample sizes for their predictions. Agrawal and Pandya(2015) used 100 students and Moucary et al.(2011) 73 students. Although the perfor?mance reported in these Neural Network studies are high, the portability of their findings is hard to determine. The network trained on 73 students might achieve good results for those students, but it might fare significantly worse if presented with new students that might come from another university or from another academic year. More students in a dataset should result in a more diverse sample and could therefore improve generalizability of the findings .

The second differentiating factor between studies is the amount of courses contained in their datasets. Courses can vary widely in length, difficulty and content. An art history course is vastly different from a course about linear algebra, and require very different study approaches. The history course might require more memorization, whichcould translate in more time spent reading slides in an LMS, while a Linear Algebra course might require more practical application, which could be done by performing more quizes and assignments. If we would use a model trained on the art history course data to predict performance for the linear algebra course students, the results might be dissapointing due to their differences. If one wants to use a trained model to predict performance on different courses, it would be wise to have trained that model on as many courses as possible to

account for possible differences between theses courses. In the studies involving student performance prediction with Neural Networks we encountered, a majority used data coming from one course or two courses .Another factor affecting performance is the difference in predictors used for classification.Predictors being used in student performance prediction vary widely across studies, often encountered predictors are age . But also more uncommon predictors have been used. For example, whether a student is in a romantic relationship or the strictness of their parents. Various predictors have different predictive qualities. Some correlate strongly to the student performance while others have only limited influence. Research has shown that the best predictor for student success is previously obtained grades, which can come from previous courses, assignments made during a course or from entry-tests (Richardson et al., 2012; Dollinger et al., 2008). If a student scored well on a test for a specific course, he will likely score well on future tests for that course.This predictor is used in all of the student performance prediction studies. However, previous grades might not always be available for various reasons. As an example we can take Massive Open Online Courses (MOOCs) where everyone can subscribe without having to enter previous academic achievements. Or previous grades are not available because it’s the first course a student is taking at a certain educational institution. Being able to perform accurate prediction,without knowing previous grades enables a more flexible usage of the predictive model.

CHAPTER-3
RESULT EVALUATION

In order to evaluate the performance and thus applicability of the various classification algorithms, the accuracy and recall metrics are used. Accuracy is the percentage of correctly classified instances among the total number of classified instances. This is a widely used evaluation metric in machine learning, which makes it a good metric to compare performance between studies. A higher accuracy means a more accurate and thus higher performing algorithm. When using the algorithm to predict if a person will require academic assistance or not, we want to minimize the number of students that are labeled as not requiring assistance, that are in fact at risk of failing the course. We do not want students that require help slipping through the system unnoticed. We can measure the propensity of the algorithm for these false negatives by measuring the recall. Recall measures how many relevant instances are successfully selected and is calculated by dividing the number of true positives by the number of true positives plus false positives. Maximizing recall will ensure that as few help requiring students go through the system unnoticed. Ideally both accuracy and recall are maximized to obtain the most suitable classification algorithm with the best parameters. To have a reference to compare the classifiers against, a majority baseline was defined. It looks at the repartition of the labels in the data and always predicts the most frequently occuring label.This allows us to get a better insight in the repartition of the labels within the data. For experiments 1 and 2, with the data for all 17 courses, the majority class is the ”does not require assistance” label; out of the 4601 students, 58.3% passed their course and thus do not require any academic assistance. This results in an accuracy of 58.3% but a recall of 0 for this majority baseline. The recall is 0 because it does not predict any ”requires assistance” labels. For experiment 3 this majority baseline is calculated for each course individually. In order to evaluate the importance of certain predictors, feature importance statistics can be generated that give an indication as to which predictor relatively contribute most to a correct prediction. These statistics are extracted from the Random Forest classifier. One of the risks during the training and parameter tuning phase is over-fitting the classifiers. This means that the parameters of the algorithm are tailored to achieve maximum classification performance on the examples in the training data, but the model does not perform well on other never seen before instances. In order to minimize the risk of over-fitting, 10 fold Cross validation was used for the training and validation phase. This technique splits the training data in 10 folds, at each iteration, nine folds are used to train the algorithm, while the left over fold is used to measure the accuracy of the classifier after training to validate their

performance. This process is then repeated for the remaining nine folds, the mean accuracy for all these folds is then calculated and gives an appropriate representation of the performance of the classifier on new data. Once the classifiers have been trained and the best parameters have been chosen, we record the accuracy and recall

CHAPTER-4
PREDICTOR CALCULATION

The data in the database cannot be used as is. Predictors that encode certain properties of the data need to be extracted first. The following predictors were extracted from the data:

- Course ID
- Number of sessions
- Total time online
- Average length of login session
- Total number of clicks
- Average number of clicks per session
- Regularity of logins (frequency)
- Longest time without activity
- Number of forum posts
- Number of messages sent
- Number of quizzes participated
- Average quiz grades
- Number of assignments participated
- Assignment grades

Each predictor (except for the number of messages) is calculated for each student/course combination, for courses taken during the 2014-2015 academic period. The data was extracted and cleaned using the R statistical programming language in RStudio. The data manipulation library DPLYRwas used to help in this task. The data itself is located in a MySQL database, which was accessed from R with the RMySQL library. It allows for SQL queries to be executed within R and the results of these queries can then be stored in an R Dataframe. After calculating and cleaning the predictors, they were stored in a Dataframe and saved as a comma-separated values (CSV) file format which can be imported to a Python environment from which the classifiers
are trained and executed. In order to prevent outliers from impacting the data and predictions in unwanted ways, the data of each course is limited to the 10 weeks the course is active. This prevents student actions, like accessing the course page later in the academic year to check assesment grades, to impact
predictors like study regularity. The first step was to delete any data that did not belong to the 2014-2015 academic year which lasted from 2014-9-1 to 2015-7-4. Because no information was available as to when each course took place, we had to determine it statistically. The academic year at the TU Eindhoven consists of four quartiles, with each course belonging to exactly

one quartile. During the 2014-2015 year, the quartiles were as follows: quartile 1: 2014-9-1 to 2014-11-8, quartile 2: 2014-11-10 to 2015-1-31, quartile 3: 2015-2-2 to 2015-4-18 and quartile 4: 2015-4-20 to 2015-7-4. For each course, the median of all the session dates was calculated. The course was then assigned to the quartile in which the median for that course occured. All course data occuring outside of the quartile in which a course took place was discarded. Each predictor was calculated as follows:

Course ID: The course ID indicates what course an action belongs to. Every action logged in tables of the database has a corresponding course ID that can directly be extracted from the courseid column or can be retrieved from a related table.

Number of Sessions, Total Time Online, Average Session Length: Most students do not log out of their Moodle sessions, they just close their browser which does not leave atrace in the logfile. This makes it harder to determine the exact end time of a session. In order to get a general idea of the length the rule that 20 minutes of inactivity (no new actions) would mean the end of a session was used. Each row in that table corresponds to an action and contains the course that action belongs to, the type of action (logging in, viewing a page, etc.), and the time at which that action occurred. The end time of a session was set as the time of the last action before the inactivity timer elapsed. Also switching from one course to another leads to the end of that session and the start of a new session for another course. The time of first activity in a certain course is used as the start time. To get the session length the start time was subtracted from the end of session time. Sessions that lasted less than 30 seconds were excluded from the list, because they may not correspond to study activities but rather to checking updates on the course page. The total time online by a student for one course is calculated by adding all the session lengths together. The average time per session was calculated by dividing the total time by the number of sessions. Some students did not use the LMS, this means that they have a total time online of 0. Rows with 0 time online were removed as they do not reflect the usage of Moodle. This resulted in the removal of 107 student rows from the data.

Total Number of Clicks, Average Clicks per Session: The click information was extracted from the mdl27 logstore standard log table by counting the number of actions of a certain user during one course. Each action, like opening the course page, viewing a document or posting a forum message were counted as a click. The average number of clicks per session was obtained by dividing the total number of clicks by the number of login sessions.

Number of Forum Posts: In order to determine what forum belonged to what course. It was then possible to count the number of posts per course and user.

Number of Messages Sent: The amount of messages sent by a certain user could be extracted from the mdl27 message table. This predictor is not tied to a specific course as messages are sent from the general Moodle interface from one user to another.

Regularity of logins: This predictor was calculated using the sessions list that was calculated for the session statistics above. A list was created with the times between sessions, the standard deviation of that list was calculated. A lower standard deviation means a higher regularity. Longest time without activity: It was determined by looking for the longest time between two sessions during an academic unit of 10 weeks (the time during which the course was active). Holiday were accounted for in the inactivity time as they might skew the inactivity time towards holiday periods. It was done by subtracting the total time of the holiday in seconds from the inactivity time when that period of inactivity occurred during a holiday.

Average Quiz Grades, Number of Quizzes Participated: The grades for quizzes were extracted from the mdl27 quiz grades table. The grades are mostly on a scale from 0 to 10, but some have more exotic scales going up to 31. The information about what scale was used and which course a quiz belonged to was extracted from the mdl27 quiz table. The number of quizzes varies from one course to another. It is therefore necessary to take the average of a students quiz grades over one course. The number of quiz grades for one course is used to determine the amount of quizzes a student has participated in, with the reasoning that if a student has a grade for a quiz, he or she has participated in said quiz. Missing quiz grades were replaced by the mean of the available quiz grades, while the missing values for number of quizes participated were replaced by 0. This resulted in the replacement of 2250 missing values for both the quiz grades and number of quizes participated.

Assignment Grades, Number of Assignments Made: The same process was applied for the extraction of the assignment grades. These are stored in the mdl27 assign grades table. For two courses, all assignment grades were set to 0, which can most likely be attributed to an error. As such, the grades for these two courses, corresponding to 18 grades, were deleted. Additionally, not all courses made use of assignments. Missing assignment predictors were replaced by the mean of the available assignment grades, and the missing values for number of assignments

made were replaced by 0. This resulted in the replacement of 4418 missing values for both grades and number of assignments made.

Exam Grade: In order to have labels that will be used for predicting, the student grades need to be extracted for each course. These can be found in the totalgrades table. This table contains the students exam grade and the final grade for the course. The final grade is weighted average of the exam grade and assignment and quiz grades. This means that it is directly dependent on the quiz and assignment grades which are used as predictors. This direct dependence is something we want to avoid. The exam grade was therefore used as the label to be predicted as it is not directly dependent on any of the other features. If a student participates in the exam, even if he or she hands in an empty answer sheet, they still receive a grade of 1.0. If the grade is equal to 0 it means that the student did not participate in the exam. All rows with exam grades that were equal to 0 or with missing grades were removed from the data, which resulted in the removal of 240 student rows. Because binary classification is performed (requires assistance or does not require assistance), the grades needed to be transformed from a numerical value (7.1/10 for example) to a categorical one that matches the intent of this study, which is to create a system to detect students that are at risk to fail a course. This is achieved by assigning a 1 (requires assistance) to all grades below 5.5 and a 0 (does not require assistance) to all grades between 5.5 and 10. Descriptive statistics for the predictors can be found below.

Predictor	Occurence in Data	Mean	SD
Number of sessions	4601	33.4	22.4
Total session length (min)	4601	715.1	519.3
Average session length (min)	4601	20.9	9.1
Number of actions	4601	660.9	663.4
Average actions per session	4601	18.4	12.1
Mean assignment grade	183	3.7	4.2
Number of assignments made	183	2.0	1.8
Mean quiz grade	2351	6.4	1.8
Number of quizes made	2351	6.9	2.6
Messagessent	56	1.8	5.1
Nr. of forum posts	69	1.8	1.6
Regularity (hours)	4601	94.9	72.1
Maximum inactivity time (hours)	4601	275.8	149.8
Grade	4601	5.8	2.0
CourseID	4601	-	-

The data contains information for 4601 students, but certain predictors are not available for all students. Some courses did not make us of quizes or assignments

for example, as such, only a limited amount of quiz and assignment predictors are available, 2351 and 183 respectively. Missing values have been replaced by either 0 or the mean of that feature column depending on the case

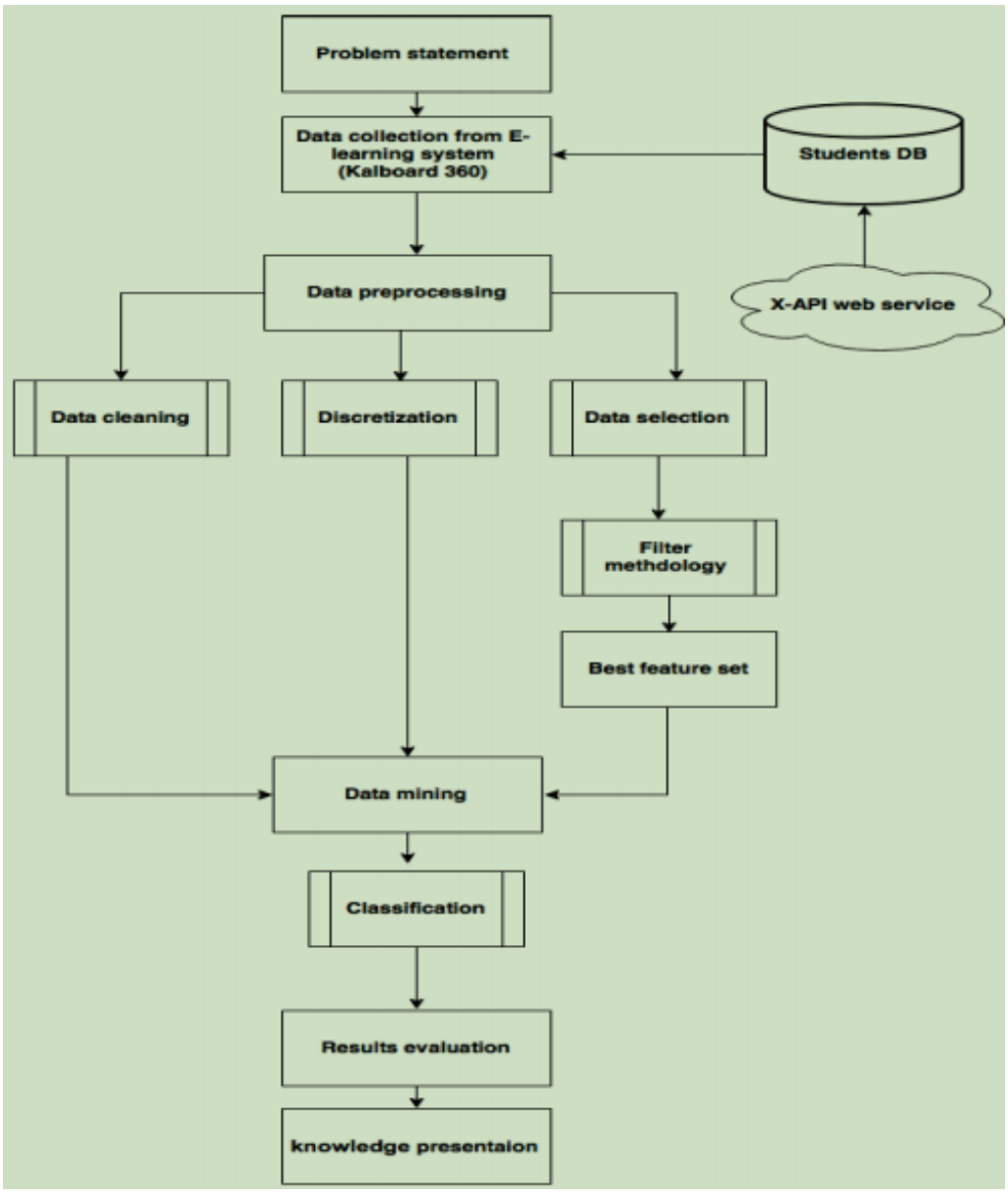
CHAPTER-5
Project Design include the diagrams

This is an educational data set which is collected from learning management system (LMS) called Kalboard 360. Kalboard 360 is a multi-agent LMS, which has been designed to facilitate learning through the use of leading-edge technology. Such system provides users with a synchronous access to educational resources from any device with Internet connection.

Feature Category	Feature	Description
Demographical Features	Nationality	Student nationality
	Gender	The gender of the student (female or male)
	Place Of Birth	Place of birth for the student (Jordan, Kuwait, Lebanon, Saudi Arabia, Iran, USA)
	Relation	Student's contact parent such as (father or mum)
Academic Background Features	Stage ID	Stage student belongs such as (Lower level , Middle level , and high level)
	Grade ID	Grade student belongs such as (G-01, G-02, G-03, G-04, G-05, G-06, G-07, G-08, G-09, G-10, G-11, G-12)
	Section ID	Section student belongs such as (A, B, C).
	Semester	School year semester such as (First or second).
	Topic	Course topic such as (Math, English, IT, Arabic, Science, Quran)
	Teacher ID	Teacher who teach this particular course.
Behavioral Features	Raised hand on class	Student Behavior during interaction with Kalboard 360 e-learning system.
	Opening resources	
	discussion groups	
	Viewing announc4ments	

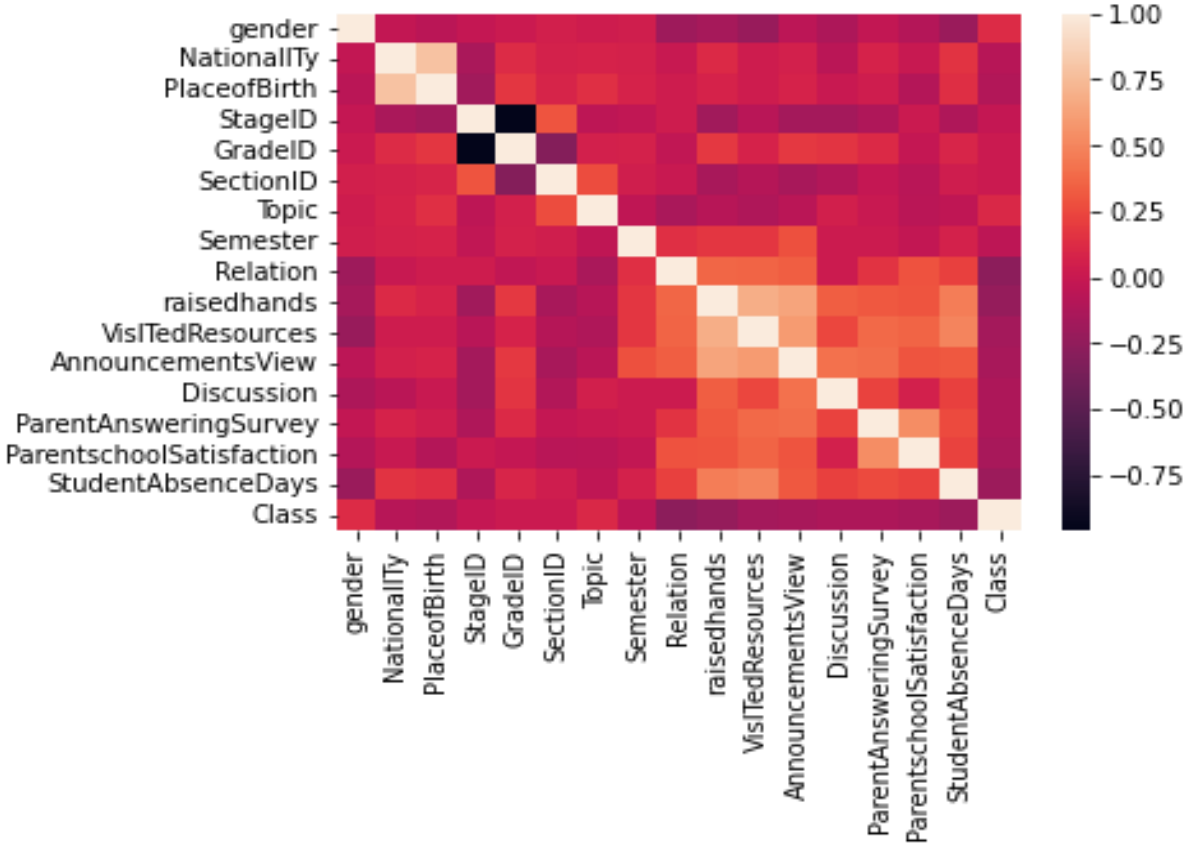
The dataset consists of 480 student records and 16 features. The features are classified into three major categories: (1) Demographic features such as gender and nationality. (2) Academic background features such as educational stage, grade Level and section. (3) Behavioral features such as raised hand on class, opening resources, answering survey by parents, and school satisfaction.

Architectural diagram of the student performance :



CHAPTER-6
RESULT

Coorelation matrix implementation:



Dataset:

Here we have 480 rows and 17 columns where we will use class attribute/feature as our target attribute.

```
[ ]  1  dataset.head()
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Discussion
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	15	16	2	20
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	20	20	3	25
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10	7	0	30
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30	25	5	35
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	40	50	12	50

Support Vector Classifier is:
Accuracy of SVC is 82%

```
[ ]  1
     2  #gamma is kernel cofficient
     3  model=SVC(kernel='linear', C=2.0, random_state=42)
     4  model.fit(x_train,y_train)
     5  y_predict_svm=model.predict(x_test)
```

```
[ ]  1  print('Misclassified samples:',(y_test != y_predict_svm).sum())
     2  print('Accuracy: %.2f'%accuracy_score(y_test, y_predict_svm))
     3  print(classification_report(y_test,y_predict_svm))
```

```
Misclassified samples: 17
Accuracy: 0.82
              precision    recall  f1-score   support

           0       0.76      0.73      0.74        22
           1       0.83      0.92      0.87        26
           2       0.85      0.81      0.83        48

    accuracy                           0.82        96
   macro avg       0.81      0.82      0.82        96
weighted avg       0.82      0.82      0.82        96
```

Decision Tree Classifier:
Accuracy of Decision Tree Classifier is 79%

```
[ ]  1  DTclassifier=DecisionTreeClassifier(random_state=42)
     2  DTclassifier.fit(x_train,y_train)
     3  y_predict_DTC=DTclassifier.predict(x_test)
```

```
[ ]  1  print('Misclassified samples:',(y_test != y_predict_DTC).sum())
     2  print('Accuracy: %.2f'%accuracy_score(y_test, y_predict_DTC))
     3  print(classification_report(y_test,y_predict_DTC))
```

```
Misclassified samples: 20
Accuracy: 0.79
              precision    recall  f1-score   support

           0       0.67      0.73      0.70        22
           1       0.88      0.88      0.88        26
           2       0.80      0.77      0.79        48

    accuracy                           0.79        96
   macro avg       0.79      0.79      0.79        96
weighted avg       0.79      0.79      0.79        96
```

Random Forest Classifier:
The Accuracy of Random Forest Classifier is 81%

```
1  forest = RandomForestClassifier(n_estimators =30,random_state=42)
2  forest.fit(x_train, y_train)
3  forest_pred = forest.predict(x_test)
```

```
1  print('Misclassified samples:',(y_test != forest_pred).sum())
2  print('Accuracy: %.2f'%accuracy_score(y_test,forest_pred))
3  print(classification_report(y_test,forest_pred))
```

```
Misclassified samples: 18
Accuracy: 0.81
              precision    recall  f1-score   support

           0       0.68      0.77      0.72        22
           1       0.86      0.96      0.91        26
           2       0.86      0.75      0.80        48

    accuracy                           0.81        96
   macro avg       0.80      0.83      0.81        96
weighted avg       0.82      0.81      0.81        96
```

Stocastic Gradient Descent:
The Accuracy of SGD is 64%

```
sgd_clf =SGDClassifier(random_state=42)
sgd_clf.fit(x_train,y_train)
sgd_pred =sgd_clf.predict(x_test)
```

```
print('Misclassified samples:',(y_test != sgd_pred).sum())
print('Accuracy: %.2f'%accuracy_score(y_test,sgd_pred))
print(classification_report(y_test,sgd_pred))
```

```
Misclassified samples: 35
Accuracy: 0.64
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        22
           1       0.84      0.62      0.71        26
           2       0.58      0.94      0.72        48

    accuracy                           0.64        96
   macro avg       0.48      0.52      0.48        96
weighted avg       0.52      0.64      0.55        96
```

CHAPTER-7
CONCLUSION

I have Worked upon SVM , Desicion tree,Random Forest Classifier,Stocastic Gradient Descent,and now I am working on building a model with high accuracy of Deep Neural Network.

In this problem, we have to build a Deep Neural Network linear classifier model to predict the performance of students. This process should be followed once the dataset is preprocessed: data cleaning and data transformation. The DNN model will be built using python3 and tensorflow 1.3.0. Artificial neural networks (ANNs) are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and discovering new knowledge. Due to their excellent capability of self?learning and self-adapting, they have been extensively studied and have been successfully utilized to tackle difficult real-world problems (Bishop 1995; Haykin 1999) and are often found to be more efficient and more accurate than other classification techniques (Lerner et al. 1999). Classification with a neural network takes place in two distinct phases. First, the network is trained on a set of paired data to determine the inputoutput mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data. Although many different models of ANNs have been proposed, the feedforward neural networks (FNNs) are the most common and widely used in a variety of applications.

Other approaches: -

- **Network-Based Clustering**
- **Baseline Methods**

CHAPTER-8
REFERENCES

1.Kumar, D. R., Krishna, T. A., & Wahi, A. (2018). Health Monitoring Framework for in Time Recognition of Pulmonary Embolism Using Internet of Things. Journal of Computational and Theoretical Nanoscience, 15(5), 1598–1602. https://doi.org/10.1166/jctn.2018.7347

2. Krishnasamy, L., Dhanaraj, R. K., Ganesh Gopal, D., Reddy Gadekallu, T., Aboudaif, M. K., &Abouel Nasr, E. (2020). A Heuristic Angular Clustering Framework for Secured Statistical Data Aggregation in Sensor Networks. Sensors, 20(17), 4937. https://doi.org/10.3390/s20174937

3. Dhiviya, S., Malathy, S., & Kumar, D. R. (2018). Internet of Things (IoT) Elements, Trends and Applications. Journal of Computational and Theoretical Nanoscience, 15(5), 1639–1643. https://doi.org/10.1166/jctn.2018.7354

4. Rajesh Kumar, D., & Shanmugam, A. (2017). A Hyper Heuristic Localization Based Cloned Node Detection Technique Using GSA Based Simulated Annealing in Sensor Networks. In Cognitive Computing for Big Data Systems Over IoT (pp. 307–335). Springer International Publishing.https://doi.org/10.1007/978-3-319-70688-7_13

5. Prasanth, T., Gunasekaran, M., & Kumar, D. R. (2018, December). Big data Applications on Health Care. 2018 4th International Conference on Computing Communication and Automation (ICCCA). 2018 4th International Conference on Computing Communication and Automation (ICCCA). https://doi.org/10.1109/ccaa.2018.8777586

6. Sathish, R., & Kumar, D. R. (2013, April). Dynamic Detection of Clone Attack in Wireless Sensor Networks. 2013 International Conference on Communication Systems and Network Technologies. 2013 International Conference on Communication Systems and Network Technologies (CSNT 2013). https://doi.org/10.1109/csnt.2013.110

7. Rajesh Kumar D, & ManjupPriya S. (2013, December). Cloud based M-Healthcare emergency using SPOC. 2013 Fifth International Conference on Advanced Computing (ICoAC). 2013 Fifth International Conference on Advanced Computing (ICoAC).https://doi.org/10.1109/icoac.2013.6921965

8. Sathish, R., & Kumar, D. R. (2013, March). Proficient algorithms for replication attack detection in Wireless Sensor Networks — A survey. 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN). 2013 International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN). https://doi.org/10.1109/ice-ccn.2013.6528465

9. Soumya Ranjan Jena, Raju Shanmugam, Rajesh Kumar Dhanaraj, Kavita Saini Recent Advancesand Future Research Directions in Edge Cloud Framework. (2019). International Journal of Engineering and Advanced Technology, 9(2), 439–444.https://doi.org/10.35940/ijeat.b3090.129219

10. Kumar, R. N., Chandran, V., Valarmathi, R. S., & Kumar, D. R. (2018). Bitstream Compression for High Speed Embedded Systems Using Separated Split Look Up Tables (LUTs). Journal of Computational and Theoretical Nanoscience, 15(5), 1719–1727. https://doi.org/10.1166/jctn.2018.7367

11. Lalitha, K., Kumar, D. R., Poongodi, C., & Arumugam, J. (2021). Healthcare Internet of Things –The Role of Communication Tools and Technologies. In Blockchain, Internet of Things, and Artificial Intelligence (pp. 331–348). Chapman and Hall/CRC.https://doi.org/10.1201/9780429352898-17

12. Rajesh Kumar, D., Rajkumar, K., Lalitha, K., & Dhanakoti, V. (2020). Bigdata in the Management of Diabetes Mellitus Treatment. In Studies in Big Data (pp. 293–324). Springer Singapore.https://doi.org/10.1007/978-981-15-4112-4_14

13. Chandraprabha, M., & Dhanaraj, R. K. (2020, November 5). Machine learning based PedanticAnalysis of Predictive Algorithms in Crop Yield Management. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). https://doi.org/10.1109/iceca49313.2020.9297544

14. Dhanaraj, R. K., Rajkumar, K., & Hariharan, U. (2020). Enterprise IoT Modeling: Supervised,Unsupervised, and Reinforcement Learning. In Business Intelligence for Enterprise Internet of Things (pp. 55–79). Springer International Publishing. https://doi.org/10.1007/978-3-030-44407-5_3

15. Cynthia, J., Sankari, M., Suguna, M., & kumar, D. R. (2018, December). Survey on Disaster Management using VANET. 2018 4th International Conference on Computing Communication and Automation (ICCCA). 2018 4th International Conference on Computing Communication and Automation (ICCCA). https://doi.org/10.1109/ccaa.2018.8777331

16. Dhanaraj, R. K., Shanmugam, A., Palanisamy, C., & Natarajan, A. (2016). Optimal Clone Attack Detection Model using an Energy-Efficient GSA based Simulated Annealing in Wireless Sensor Networks. Asian Journal of Research in Social Sciences and Humanities, 6(11), 201. https://doi.org/10.5958/2249-7315.2016.01186.2

17. Sathya, K., & Kumar, D. R. (2012, February). Energy efficient clustering in sensor networks using Cluster Manager. 2012 International Conference on Computing, Communication and Applications. 2012 International Conference on Computing, Communication and Applications (ICCCA). https://doi.org/10.1109/iccca.2012.6179177

18. Lalitha, K., Varadhaganapathy, S., Santhoshi, S., & Kumar, D. R. (2018, December). A Review on Possibilities of Hearing Loss and Implantable Hearing Devices for Teenagers. 2018 4th International Conference on Computing Communication and Automation (ICCCA). 2018 4th International Conference on Computing Communication and Automation (ICCCA). https://doi.org/10.1109/ccaa.2018.8777336

18. Krishnamoorthi, S., Jayapaul, P., Dhanaraj, R.K. et al. Design of pseudo-random number generator from turbulence padded chaotic map. Nonlinear Dyn (2021). https://doi.org/10.1007/s11071-021-06346-x

19. R. K. Dhanaraj, L. Krishnasamy, O. Geman and D. R. Izdrui, "Black hole and sink hole attack detection in wireless body area networks," Computers, Materials & Continua, vol. 68, no.2, pp. 1949–1965, 2021. doi:10.32604/cmc.2021.015363

20. Rajesh Kumar Dhanaraj, Lalitha, K., Anitha, S., Khaitan, S., Gupta, P., & Goyal, M. K. (2021).Hybrid and dynamic clustering-based data aggregation and routing for wireless sensor networks[JB]. Journal of Intelligent & Fuzzy Systems, 1–15.

21. M. D. Ramasamy, K. Periasamy, L. Krishnasamy, R. K. Dhanaraj, S. Kadry and Y. Nam, "Multi-Disease Classification Model using Strassen’s Half of Threshold (SHoT) Training Algorithm in Healthcare Sector," in IEEE Access, doi: 10.1109/ACCESS.2021.3103746.

22. Ramakrishnan, V., Chenniappan, P., Dhanaraj, R. K., Hsu, C. H., Xiao, Y., &; Al-Turjman, F.(2021). Bootstrap aggregative mean shift clustering for big data anti-pattern detection analytics in 5G/6G communication networks. Computers & Electrical Engineering, 95, 107380.

23. KRISHNASAMY, L., RAMASAMY, T., DHANARAJ, R., & CHINNASAMY, P. (2021). A geodesic deployment and radial shaped clustering (RSC) algorithm with statistical aggregation in sensor networks. Turkish Journal of Electrical Engineering & Computer Sciences, 29(3).