A Project Report

on

Diabetes Prediction Model Using Deep Learning

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

Bachelor of Technology in Computer Science

Under The Supervision of
Name of Supervisor:
Mr.Ravi Sharma

Submitted By

Prince Gupta 19SCSE1010762
Shashi Kiran 19SCSE1010605

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
October, 2021-22

SCHOOL OF COMPUTING SCIENCE AND GALGOTIAS UNIVERSITY, GREATER NOIDA ENGINEERING

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Diabetes Prediction Model Using Deep Learning"** in partial fulfillment of the requirements for the award of the Bachelor's 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 month, Year to Month and Year, under the supervision of Mr. Ravi Sharma Associate Professor, 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 us for the award of any other degree of this or any other places.

Prince Gupta(19SCSE1010762)

Shashi Kiran(19SCSE1010471)

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

Ravi Sharma

Assistant Professor

Department of Computer Science and Engineering

CERTIFICATE

The Final Project Viva-Voce examination of Prince Gupta(19SCSE1010762) Shashi Kiran(19SCSE1010605) has been held on _____ and his/her work is recommended for the award of Bachelor's of Technology.

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

Signature of Project Coordinator Signature of Dean

Date: November, 2013

Place: Greater Noida

Abstract

Diabetes Mellitus is among critical diseases and lots of people are suffering from this disease. Age, obesity, lack of exercise, hereditary diabetes, living style, bad diet, high blood pressure, etc. can cause Diabetes Mellitus. People having diabetes have high risk of diseases like heart disease, kidney disease, stroke, eye problem, nerve damage, etc. Current practice in hospital is to collect required information for diabetes diagnosis through various tests and appropriate treatment is provided based on diagnosis. Machine learning plays an significant role in healthcare industries. Healthcare industries have large volume databases. Using machine learning one can study huge datasets and find hidden information, hidden patterns to discover knowledge from the data and predict outcomes accordingly. In existing method, the classification and prediction accuracy is not so high. In this paper, we have proposed a diabetes prediction model for better classification of diabetes which includes few external factors responsible for diabetes along with regular factors like Glucose, BMI, Age, Insulin, etc. Classification accuracy is boosted with new data-set compared to existing data-set. Further with imposed a pipeline model for diabetes prediction intended towards improving the accuracy of classification.

Keyword : Diabetes, Deep Learning, Prediction, Dataset.

Table of Contents

Chapter-1
Introduction

1.1 Introduction:

Diabetes can be considered as one of the main challenges in the healthcare community worldwide and its impact is increasing at a very high pace. Consequently, it is the seventh major reason for the premature death rate in 2016 worldwide mentioned by the World Health organization (WHO) [1].According to the diabetes global pervasiveness, 1.6 million got died each year because of diabetes [2]. WHO has demonstrated in its first global report that the number of persons suffering from diabetes increased from 108 million (4.2%) to 422 million (8.5%) till the end of 2014 [3]. On world diabetes day 2018, WHO has joined the partners from all over the world for showcasing the impact of diabetes. According to WHO, 1 in 3 adult is reported overweight and the problem is increasing day by day. Diabetes can be considered as a chronic disease in which glucose (blood sugar) is not metabolized in the body (glucose is produced from the food we eat); therefore, it increases the level of sugar in the blood over the acceptable limits. Healthcare Industry collects a huge amount of data related to healthcare,but that data is unable to perceive undetected patterns for making effective decisions.Various undetected errors and hidden patterns give rise to diverse data mining and machine algorithms which can draw efficient results with reliable accuracy. This research work represents comprehensive studies done on the PIMA datasets using data mining algorithms like DT, NB,ANN, and DL .The comparison of algorithms is represent-ed in a logical and well-organized manner from which DL provides more effective and prominent results. DL is a technology that self-learns from data and is used effectively for predicting diabetes nowadays [4]. A DL network is a technique that uses ANN properties in which neurons are inter-connected to each other with lot of representation layers

[4,6,10]. DL learns the representation of data by enlarging the level of consideration from one layer to another hence increasing the accuracy. The model achieves high accuracy of 98.07% by employing DL in Rapid Miner tool which proposes a well-structured diabetes knowledge formatted for medical officials and practitioners. Moreover, the task is to reduce the efforts and to provide better results in comparison with the traditional methods . These machine learning method to improve the accuracy of the available methods. But DL and ANN provide the best results as they are more reliable, robust and accurate in terms of prediction of the disease.The remaining part of the paper is organized in the following manner: third section puts forth the previous important work done on diabetes prediction using data mining algorithms. Fourth section of the paper presents the data-set description,and proposed methodology. Fifth section covers the results and discussion part. The paper concludes in sixth section along with future scope.

1.2 Problem Formulation:

As Diabetes continues to spread across the world, one of the greatest challenges to response efforts is a lack of data and evidence about causes and case fatality. We will learn more about the Diabetes increment and the populations most at risk as different treatments arrive in the hands of physicians. Ideally, this information will be used to target response efforts at populations most in need, guide policy decisions, inform the donor community, and support decision-making by organizations whose work spans multiple countries.

1.3 Relevence of Project

In recent years, using Machine Learning has been used with increasing frequency to predict the possibility of disease. Many algorithms and toolkits have been created and studied by researchers. These have highlighted the tremendous potential of this research field.

Based on several studies, we found that a commonly used dataset was the Pima Indians Diabetes Dataset from the University of California, Irvine (UCI) Machine Learning Database.

1.4 Scope of the Project

The early intervention of diabetes can reduce the prevalence of diabetes and hence the economic burden due to it. Machine Learning techniques play an important role in treatment plan workout, rehabilitation, chronic diseases management plan etc. Long term follow up plan may be easily guided and keen supervision is possible. The systems may definitely helpful in reduction of cost of patient management by avoiding unnecessary investigations and patients follow up. These prediction systems will add accuracy and time management. Computer-based patient support systems benefit patients by providing informational support that increases their participation in health care.

Chapter-2

Literature Survey

Proposed random Forest algorithm for the Prediction of diabetes develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by using Random Forest algorithm in machine learning technique. The proposed model gives the best results for diabetic prediction and the result showed that the prediction system is capable of predicting the diabetes disease effectively, efficiently and most importantly, instantly. Nonso Nnamoko etal[6]. presented predicting diabetes onset: an ensemble supervised learning approach they used five widely used classifiers are employed for the ensembles and a meta-classifier is used to aggregate their outputs. The results are presented and compared with similar studies that used the same dataset within the literature. It is shown that by using the proposed method, diabetes onset prediction can be done with higher accuracy. Tejas N. Joshi et al.[7] presented Diabetes Prediction Using Machine Learning Techniques aims to predict diabetes via three different supervised machine learning methods including: SVM, Logistic regression, ANN. This project proposes an effective technique for earlier detection of the diabetes disease. Deeraj Shetty et al.[8] proposed diabetes disease prediction using data mining assemble Intelligent Diabetes Disease Prediction System that gives analysis of diabetes malady utilizing diabetes patient's database.In this system, they propose the use of algorithms like Bayesian and KNN (K-Nearest Neighbor) to apply on diabetes patient's database and analyze them by taking various attributes of diabetes for

prediction of diabetes disease. Muhammad Azeem Sarwar et al.[5] proposed study on prediction of diabetes using machine learning algorithms in healthcare they applied six different machine learning algorithms Performance and accuracy of the

applied algorithms is discussed and compared. Comparison of the different machine learning techniques used in this study reveals which algorithm is best suited for prediction of diabetes. Diabetes Prediction is becoming the area of interest for researchers in order to train the program to identify the patient are diabetic or not by applying proper classifier on the dataset.

Chapter-3

Project Design

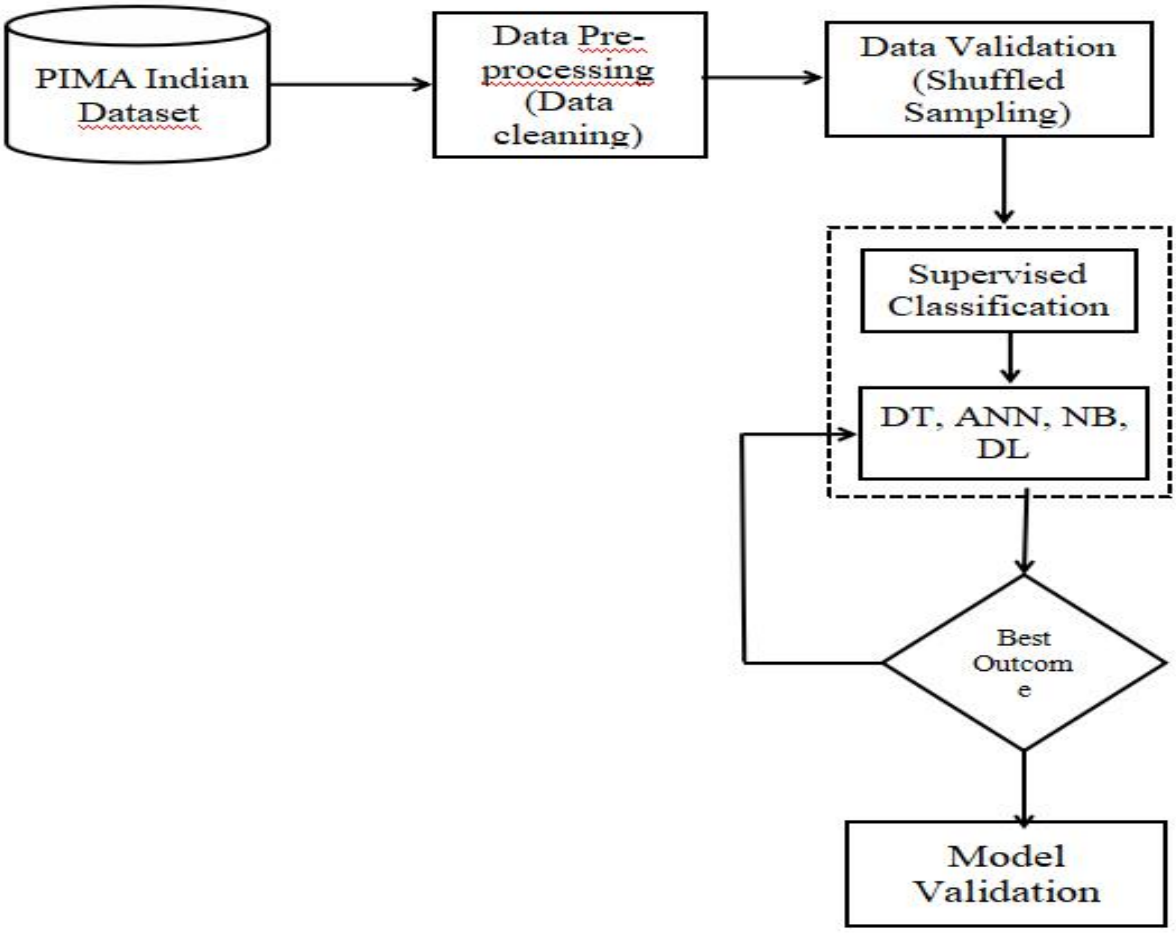

Diagram 1. System architecture for diabetes prediction model

The Diagram 1 shows the architecture of Diabetes Prediction System. The data set is taken from the PIMA. The training data set is fed to the system as input which will be initially pre processed. Data pre processing is the phase where the raw data will be transformed into meaningful and understandable format. We will test the data on different Algorithms and we will consider the algorithm providing the highest accuracy of the model.

The system initially considers the different parameters like Glucose level of an individual, BMI, age of the person, Blood pressure, insulin etc. Based on the value of Glucose level it categorizes the person whether the person is diabetic or non diabetic.

Chapter-4

Project Implementation

4.1 Module1-(Deep learning):

Deep Learning is a multi-layer feed-forward perceptron based model which also facilitates the properties of ANN and trained with stochastic gradient descent using back-propagation.The network is a collection of four layers emulating nodes and neurons, directed in uni-direction (one-way connection). Each node is connected to the next node in a single way connection and contains two hidden layers where each node trains a copy of global model parameters by applying its local data.

4.2 Module2-(Decision tree) :

DT is a graph which is used in decision analysis and demonstrate outcome as a splitting rule for every specific attribute. It is a branching graph which can be applied as visually and explicitly for decision-making outcome. Every attribute is considered as a branching node and constructs a rule at the end of the branch that divides values belonging to different classes. It is a tree-like structure as its name suggests and concludes some decision at the end that is called the leaf of the tree. The root is the most potential attribute which can be applied for prediction of the outcome of rule formation.

4.3 Module3-(Artificial neural network):

ANN is another technique for classification which is a machine learning algorithm and provides more accurate results in comparison with the existing algorithms. It is a mathematical model that is inspired by the functioning and structure of biological neurons. A neural network is a connection of multiple neurons connected as the human brain is a connection of 86 billion biological neurons. The functional connectivity in artificial neurons is mesh connectivity and each neuron has equal weight.

4.4 **Pandas:** It is an open-source, library in Python Language. Pandas provide high performance, fast, easy to use data structures and data analysis tools for manipulating numeric data and time series. Pandas is built on the numpy library and written in languages like Python, Cython, and *C*. In pandas, we can import data from various file formats like JSON, SQL, Microsoft Excel, etc.

Using Pandas, a developer can perform the following operations −

When we have to work on Tabular data, we prefer the pandas module.

Pandas has a better performance when number of rows are more.
Pandas offers 2d table object called DataFrame.

4.5 Steps involved in predicting the model functioning:-

#Import Standard Libraries

```
import pandas as pd                    #pandas-mathematical operations(dataframe)
import matplotlib.pyplot as plt        #matplotlib-2D graphs
import seaborn as sns                  #seaborn-visualization
```

#Load Dataset

```
df=pd.read_csv(r'diabetes.csv')        #dataframe-a type of array
df (Table will be formed)
df.shape                               #rows and columns
df.size                                #size of the data
df.info()                              #datatype & null values
df.isnull().sum()                      #nullvalues and returns sum
```

#Show the information on fields

```
df.describe()                          #in form of percentile
df.describe().T                        #row-->column & vice versa
```

#columns values count

```
df['Outcome'].value_counts()
#Show distribution of various columns
df.boxplot(figsize=(13,7))
plt.show()
```

```
plt.figure(figsize=(13,7))

sns.boxplot(data=df,orient='h')

plt.show()
```

#Split dataset into features and target variable

```
X=df.drop('Outcome',axis=1)

Y=df['Outcome']
```

#For replacing null value in X

```
X.replace(to_replace=0,value=X.mean(),inplace=True)

X.describe()
```

#import libraries for preparing model

```
from sklearn.model_selection import train_test_split
```

#Split X and Y into training and testing sets

```
X_train , X_test , Y_train ,
Y_test=train_test_split(X,Y,test_size=0.25,random_state=None)
```

#import library for logistic regression

```
from sklearn.linear_model import LogisticRegression
```

#instantiate the model (using the default parameters)

```
lr=LogisticRegression()
```

#fit the model with data

lr.fit(X_train_std,Y_train)

#predict values

Y_pred=lr.predict(X_test_std)

#For checking accuracy

from sklearn.metrics import accuracy_score

accuracy_score(Y_test,Y_pred)

accuracy_score(Y_test,Y_pred)*100

#import library for decision tree

from sklearn.tree import DecisionTreeClassifier

#instantiate the model (using the default parameters)

dt=DecisionTreeClassifier()

#fit the model with data

dt.fit(X_train_std,Y_train)

#predict values

Y_pred=dt.predict(X_test_std)

#For checking accuracy

```
accuracy_score(Y_test,Y_pred)
```

```
accuracy_score(Y_test,Y_pred)*100
```

#import library for MLP

```
from sklearn.neural_network import MLPClassifier
```

#fit the model

```
lp.fit(X_train_std,Y_train)
```

#predict values

```
Y_pred=mlp.predict(X_test_std)
```

#For checking accuracy

```
accuracy_score(Y_test,Y_pred)
```

```
accuracy_score(Y_test,Y_pred)*100
```

4.5 Training

Data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
627	3.845052	132.0	78.0	20.536458	79.799479	32.4	0.393	21
152	9.000000	156.0	86.0	28.000000	155.000000	34.3	1.189	42
592	3.000000	132.0	80.0	20.536458	79.799479	34.4	0.402	44
671	1.000000	99.0	58.0	10.000000	79.799479	25.4	0.551	21
464	10.000000	115.0	98.0	20.536458	79.799479	24.0	1.022	34
...	...	...	...	...	...	...	...	...
325	1.000000	157.0	72.0	21.000000	168.000000	25.6	0.123	24
335	3.845052	165.0	76.0	43.000000	255.000000	47.9	0.259	26
273	1.000000	71.0	78.0	50.000000	45.000000	33.2	0.422	21
436	12.000000	140.0	85.0	33.000000	79.799479	37.4	0.244	41
670	6.000000	165.0	68.0	26.000000	168.000000	33.6	0.631	49

576 rows × 8 columns

Diagram 2- taring dataset

Diagram 2 shows the dataset used to predict the diabetes. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

CHAPTER 5

Testing

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate oneor more properties of interest.

5.1 Testing and Validations

Validation is a complex process with many possible variations and options, so specifics vary from database to database, but the general outline is:

- Requirement Gathering
    - The Sponsor decides what the database is required to do based on regulations, company needs, and any other important factors.
    - The requirements are documented and approved.
- System Testing
    - Procedures to test the requirements are created and documented.

    - The version of the database that will be used for validation is set up.
    - The Sponsor approves the test procedures.
    - The tests are performed and documented.
    - Any needed changes are made. This may require another, shorterround of testing and documentation.
- System Release
    - The validation documentation is finalized.
    - The database is put into production.

5.2 Testing Levels

5.2.1 Functional Testing:

This type of testing is done against the functional requirements of the project. Types:

Unit testing: Each unit /module of the project is individually tested to check for bugs. If any bugs found by the testing team, it is reported to the developer for fixing.

Integration testing: All the units are now integrated as one single unit and checked for bugs. This also checks if all the modules are working properly with each other.

System testing: This testing checks for operating system compatibility. It includes both functional and non functional requirements.

Sanity testing: It ensures change in the code doesn't affect the working of the project.

Smoke testing: this type of testing is a set of small tests designed for each build. Interface testing: Testing of the interface and its proper functioning.

Regression testing: Testing the software repetitively when a new requirement isadded, when bug fixed etc.

Beta/Acceptance testing: User level testing to obtain user feedback on the product.

5.2.2 Non-Functional Testing:

This type of testing is mainly concerned with the non-functional requirements suchas performance of the system under various scenarios.

Performance testing: Checks for speed, stability and reliability of the software,hardware or even the network of the system under test.

Compatibility testing: This type of testing checks for compatibility of the system withdifferent operating systems, different networks etc.

Localization testing: This checks for the localized version of the product mainlyconcerned with UI.

Security testing: Checks if the software has vulnerabilities and if any, fix them. Reliability testing: Checks for the reliability of the software

Stress testing: This testing checks the performance of the system when it is exposedto different stress levels.

Usability testing: Type of testing checks the easily the software is being used by thecustomers.

Compliance testing: Type of testing to determine the compliance of a system withinternal or external standards.

- **Reliability**

  The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

- **Maintainability**

The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.

- **Performance**

  The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.

- **Portability**

  The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.

- **Scalability**

  The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.

- **Flexibility**

  Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure.

5.3 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

It is one of two parts of the **"Box Testing" approach** to software testing. Its counterpart, **Blackbox testing**, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings ofan application and revolves around internal testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

5.4 Different Stages of Testing

5.4.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Unit Test Plan:

- Unit Test Plan
  - Prepare
  - Review
  - Rework
  - Baseline
- Unit Test Cases/Scripts
  - Prepare
  - Review
  - Rework

Baselene

-

UnitTest

- Perform

Benefits

- Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
- Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy 'developer test' (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than 'developer tests'. Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the

cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.

- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span ofseveral days/weeks/months need to be scanned.

5.4.2 Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

- **integration testing:** Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.
- **component integration testing:** Testing performed to expose defects in the interfaces and interaction between integrated components.

- **system integration testing:** Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Tasks

- Integration Test Plan
  - Prepare
  - Review
  - Rework
  - Baseline
- Integration Test Cases/Scripts
  - Prepare

    Review

    Rework
  - Baselin

e • Integration

Test

System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. The process of testing an integrated system to verify that it meets specified requirements.

Acceptance Testing

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Chapter-6

Conclusion and Future Work

This paper aimed to implement a prediction model for the risk measurement of diabetes. As discussed earlier, a large part of the human population is in the hold of diabetes disease. If remains untreated, then it will create a huge risk for the world.Therefore In our proposed research, we have put into practice diverse classifiers on the PIMA data-set and proved that data mining and machine learning algorithm can reduce the risk factors and improve the outcome in terms of efficiency and accuracy. The outcome achieved on the PIMA Indian data is higher than other proposed methodologies on the same data-set using data mining algorithms as discussed.Accuracy achieved by the four classifiers (DT, ANN, NB, and DL) lies within the range 90–98% which is considerably high than available methods. Among the four proposed classifiers,DL is considered as the most efficient and promising for analyzing diabetes with an accuracy rate of 98.07%. In the future,we intend to develop a robust system in the form of an app or a website that can use the proposed DL algorithm to help healthcare specialists in the early detection of diabetes.

Chapter 7

Result

In this research work, outcomes were achieved by applying four classification algorithms (DL, ANN, NB, and DL) to display maximize accuracy in diabetes prediction. From these four classifiers, DL and DT provide promising accuracy(77.08%) which can be proven as a prominent tool for the prediction of diabetes at an early stage. In our proposed system we use the PIMA data-set and apply it on a DL approach.Further, it can help the healthcare practitioner and can be the second estimation for the betterment of decisions depending on extracted features. Many researchers have been previously worked on the PIMA data-set with a diverse algorithm to predict diabetes. Thus some of the researcher's work has been represented with their applied methods and achieved accuracy.

Chapter 8

References

1. “Gl0bal Rep0rt 0n Diabetes, 2016”. Available at: https://apps.wh0.int/iris/bitstream/handle/10665/204871/9789241565257_eng.pdf;jsessi0nid=2BC28035503CFAFF295E70CFB4A0E1DF?Sequence=1.

2. “Diabetes: Asia's 'silent killer'”, N0vember 14, 2013”. Available at: www.bbc.c0m/news/w0rld-asia24740288.

3. Emerging T, Fact0rs R. Diabetes mellitus , fasting bl00d gluc0se c0ncentrati0n , and risk 0f vascular disease : a c0llab0rative meta-analysis 0f 102 pr0spective studies. The Lancet. 2010;375(9733):2215–22. https://d0i.0rg/10.1016/S0140-6736(10)60484-9Elsevier Ltd.

4. Swapna G, Vinayakumar R, S0man KP. Diabetes detecti0n usingdeep learning alg0rithms. ICT Express. 2018;4(4):243–6. https://d0i.0rg/10.1016/j.icte.2018.10.00.5.Elsevier B.V.

5. “Global Report on Diabetes, 2016”. Available at: https://apps.who.int/iris/bitstream/handle/10665/204871/9789241565257_eng.pdf;jsessio nid=2BC28035503CFAFF295E70CFB4A0E1DF?Sequence=1.

6. “Diabetes: Asia's 'silent killer'”, November 14, 2013”. Available www.bbc.com/news/world-asia-24740288.

7. Mathers CD, Loncar D. Projections of global mortality and burden of disease from 2002 to 2030. 2015;3(11). https://doi.org/10.1371/journal.pmed.0030442.

8. Swapna G, Vinayakumar R, Soman KP. Diabetes detection using deep learning algorithms. ICT Express. 2018;4(4):243 https://doi.org/10.1016/j.icte.2018.10.005. Elsevier B.V.

9 . Emerging T, Fact0rs R. Diabetes mellitus , fasting bl00d gluc0se c0ncentrati0n , and risk 0f vascular disease : a c0llab0rative meta-analysis 0f 102 pr0spective studies. The Lancet. 2010;375(9733):2215–22. https://d0i.0rg/10.1016/S0140-6736(10)60484-9Elsevier Ltd.