DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

STUDENT BOARD SCORE PREDICTION : AN IMPLEMENTATION OF NEURAL NETWORK

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Sunil Shrestha / Aashish Bikram Lamichhane

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by SUNIL SHRESTHA and AASHISH BIKRAM LAMICHHANE entitled **"STUDENT BOARD SCORE PREDICTION : AN IMPLEMENTATION OF NEURAL NETWORK"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Sarbin Sayami

Assistant Professor

Institute of Science and Technology

Tribhuvan University

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by SUNIL SHRESTHA and AASHISH BIKRAM LAMICHHANE entitled **“STUDENT BOARD SCORE PREDICTION : AN IMPLEMENTATION OF NEURAL NETWORK”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………………………… Sarbin Sayami [Supervisor] Assistant Professor IOST, Tribhuvan University	………………………………………… Hitesh Karki Chief Academic Officer DWIT College
………………………………………….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	………………………………………….. Rituraj Lamsal [Internal Examiner] Lecturer DWIT College

ACKNOWLEDGEMENT

First of all, I would like to express my deepest gratitude to my supervisor Asst. Prof. Sarbin Sayami, IOST, TU for his motivation, guidance, advice, and valuable time.

I would like to acknowledge the generous help and support from Mr. Bijaya Shrestha for providing the necessary data and guidance to normalize the data to undertake this project. I would also like to thank Mr. Hitesh Karki, Chief Academic Officer, DWIT College for guidance on formatting the document and support for preparing the document.

Last but not the least, I wish to express my sincere thanks to all my friends for supporting me while training the data set and developing the application.

Aashish Bikram Lamichhane

TU Exam Roll no: 1794/069

Sunil Shrestha

TU Exam Roll no: 1819/069

Tribhuvan University
Institute of Science and Technology

STUDENT'S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

....
Aashish Bikram Lamichhane
TU Exam Roll no: 1794/069

....
Sunil Shrestha
TU Exam Roll no: 1819/069

Date: ....

ABSTRACT

Prediction is one of the powerful technique that is used in neural network for accurate prediction using back propagation technique and multilayer perceptron. A study was conducted to predict the board score of the students studying in any particular batch in DWIT college. Midterm score, pre-board score, assignment score, internal score, and attendance score of the students were used for prediction and the result shows that board score can be predicted with 95 percent accuracy.

Keywords: Neural network, back propagation, multi-layer perceptron

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATION AND ACRONYMS

DWIT	Deerwalk Institute of Technology
RMSE	Root Mean Square Error
MLP	Multi-layer perceptron
CASE	Computer-Aided Software Engineering
UML	Unified Modeling Language
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheet
JSP	Java Server Page

CHAPTER 1 : INTRODUCTION

1.1 Background

Maintaining quality of education is one of the key factor or challenge that any academic institution must consider for its long term sustainability. Maintaining the quality of education plays key role in any academic institution in this competitive world. An institution must consider several factors for maintaining the academic performance of students. Quality of education provided by any institution can directly be linked with the academic performance performed by students of that institution. There are several factors that determine the performance of students. Students as well as the college administrator are curious about the performance of students in the board exam. Moreover, prior knowledge regarding the performance of students in coming board examination can help maintain standard or quality in education in any institution. Information beforehand regarding the weak students can help allocate necessary time to them in taking proper action to enhance or boost their performance. From the analysis of the expected outcome of the students beforehand, it can help improve the quality of education in any institution as it helps in making proper decision.

There is a tendency that any academic institution is judged by the outcome of the board exam in our country. An institution as well as students are more focused on board exam. Moreover, an individual student is judged by the score he gets in board exam and his academic career is defined by the score he gets. In such a situation, a role of a tool or software that can predict the students performance or board score is incomparable both for the students as well as academic institution. College Administration can take necessary steps for maintaining the quality education beforehand if such tools or software are available. Moreover, such application can help them to predict what the student performance be like in board exam and based on the outcome of prediction concerned authority can take necessary and major steps to improve the performance of students. Based on some available data of students, it can help students to check how well they can do in board exam.

1.2 Problem Statement

The lack of knowledge to acknowledge how well the students will perform in upcoming board exam can bring a serious consequence in any institution. Today, some of the academic institutions fail because they ignore the fact that it is necessary to analyze how well their students can perform in upcoming board exam.

1.3 Objectives

1. To predict the board score of the students.
2. To provide recommendation on the basis of prediction.

1.4 Scope

The system can be used by students of DWIT college for assuring their performance in academic levels. This will help them in framing and revising their activities.

1.5 Limitation

1. Average score were used for some of the missing values.
2. Environmental factors and psychological factors are not included for prediction.

1.6 Outline of Document

The remaining part of the document is organized and represented in project block diagram as in Figure 1 given below:

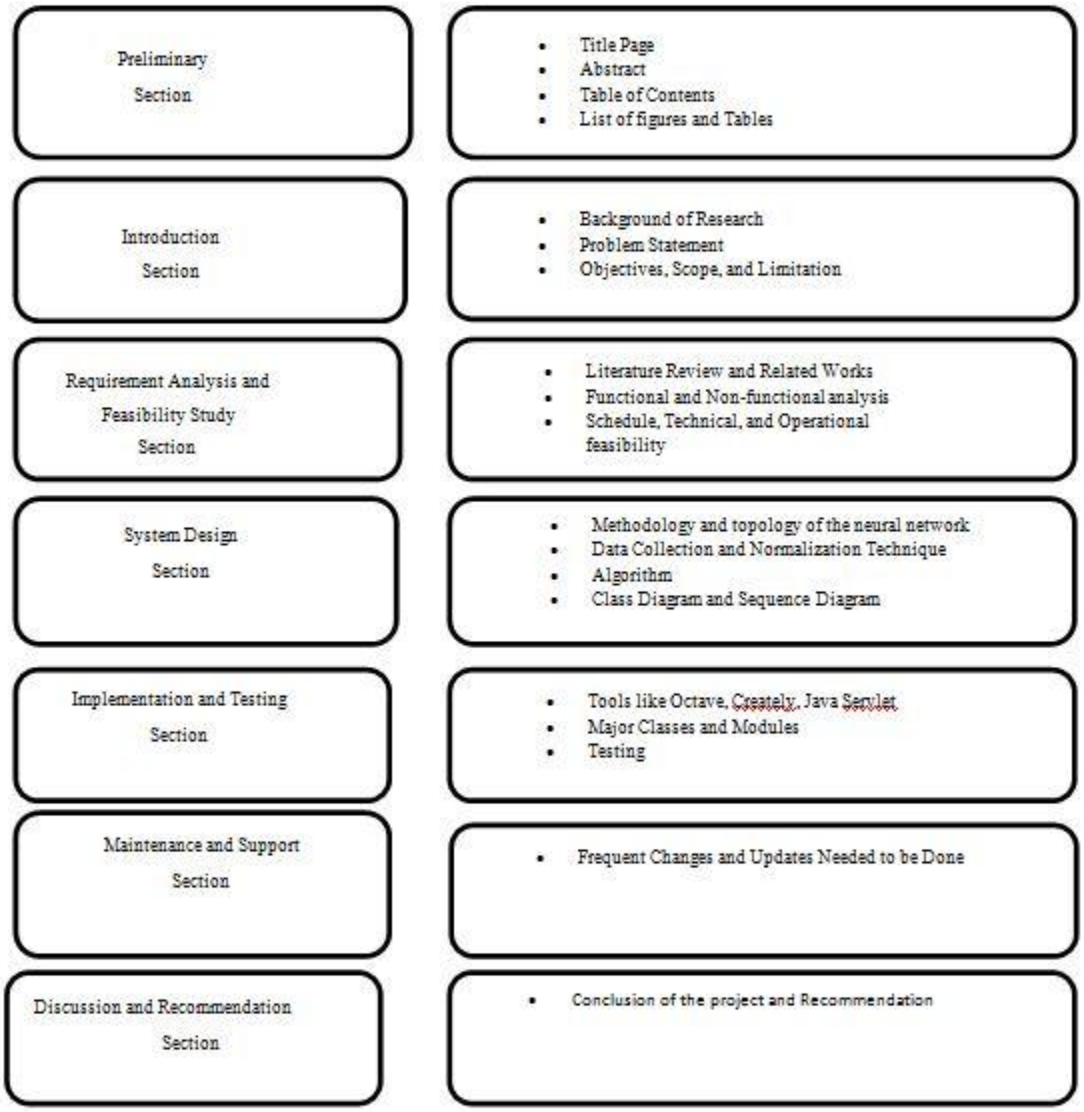


Figure 1 - Project block diagram

CHAPTER 2 : REQUIREMENT ANALYSIS AND FEASIBILITY

2.1 Literature Review

There are various factors like personal, psychological, and other socio-environmental variables which are the key components for determining an academic performance of the students. Classification, Clustering, and regression methods are some of the techniques used as a predictive model. There are several issues that are considered while performing the prediction such as predictor variable selection from academic, socio-economic and other environmental factors for effective model determination.

2.1.1 Prediction using multilayer perceptron in neural network

An experiment was performed to predict student's performance. Neural Network Topology was built based on Multilayer Perceptron with feed forward networks and trained with back propagation. The topology has two hidden layers and five processing elements in each layers and training was proceeded for 1000 iterations. Total of 112 students records were used among which 56% students records were used for training, 30% for testing and 14% for cross validation. Different factors like UME score, O/level results, Further math, Age of entry, Time before admission, educated parents, zone of secondary school attended, type of secondary school, location of school, and gender are considered and output as good, average, and poor are used. After testing, it was found that network was able to predict accurately 9 out of 11 for good data (class I), 8 out of 15 average data(class II), and 7 out of 8 poor data (class III). This achieved overall of 74% accuracy which shows the potential of ANN as effective tool for prediction (Oladokun, 2008).

2.1.2 Comparison of prediction between neural network approach and logistic regression analysis

A comparative study for prediction using Neural Network approach and logistic regression analysis was performed where data were collected from 3 different universities preceding the advanced sampling approach. Different factors like General Mathematics, Pure Mathematics, Analysis I, Analysis II, Geometry, and Linear Algebra-I were taken as input and Analysis3, Special Teaching Methods 2, Elementary Number Theory, Algebra, Problem Solving, and their success at entering a postgraduate program were chosen as output nodes. The network topology consisted 6 nodes at input layer, one hidden layer that consists 8 nodes, and output layer that comprise 6 nodes. A logarithmic sigmoid function was used as activation function between input layer and hidden layer while linear activation function used between hidden layer and output layer, and back propagation algorithm used for learning. Information was collected from 220 students while 80% was used for training and 20% data used for testing iterations was set for 10000. Neural Network could give best result over Logistic regression analysis with accurate classification success of network about 93.02% (Bahadır, 2016).

2.1.3 Comparison between neural network, decision tree, k-neighbor in prediction

An experiment was performed considering personal data, pre-university data, University data that include 11 attributes like gender, age, PlacePrevEdu, ProfilePrevEdu, ScorePrevEdu, Admission year, Admission exam year, Admission exam score, UnivSpecialtyName, Current Semester, and NumFailures among which only three attributes are numeric. Student performance prediction was performed with different algorithms like Neural Network, Decision tree, K-Nearest Neighbor, Rule Learner. The open source software WEKA is used for data mining tool for research implementation. Neural Network algorithm was found out to be the best approach for prediction of student performance and it achieved 73.59% accuracy (Kabakchieva, 2012).

2.1.4 Percentage cover in prediction of student's performance

The factors affecting student's performance in intermediate examination was explored where the data are collected from the survey of students from private colleges. The result of R square value obtained to be 0.24 which could address only 24% of the total students performance and rest 76% is explained by other factors not mentioned in the applied model (Syed Tahir Hijazi, 2014).

2.1.5 Prediction using multilayer perceptron

An experiment was performed about a significant problem in higher education is the poor results of student after student. Most of the student leave university after first year which decrease the quality of education and officers in Romania. So they used multilayer perceptron to predict the academic performance. The input variables taken into consideration are type of study program: distance education (part time) or full time education, gender of student, High-school graduation GPA, age of the student and difference in years from the moment the student graduates high-school until he/she enrolls at university. They used 1000 students from last three graduates generations from “Nicolae Titulescu” University from Bucharest(Most them left university in first year). The neural network have two hidden layer (50 neurons and 400 neurons respectively) and output layer has three neurons. They use iRPROP as an algorithm. The MSE obtained after training the network was 1.7%. The mean square error for the test data set was 1.91%. In the data set 30.1% of students belong to the class “POOR RESULTS”, 50.9% of them to the class “MEDIUM RESULTS” and 19% to the class “GOOD RESULTS” (Bogdan Oancea, 2016).

2.2 Requirement Analysis

The functional and non-functional requirements of this system is tabulated in Table 1 below:

Table 1 - Functional and Non-functional Requirement

2.2.1 Functional requirement	**2.2.2 Non-functional requirement**
Predict the board score	Read the input parameter provided by user and process the data with matrix multiplication and pass the value though activation function in each layer of network and finally display the result as the predicted score.
Train to adjust the weights of neural network	Read the parameters from training data set and process with randomly initialized weights and in each iteration compare the predicted value with target value and continue the process until it meets the target value.
Test the model for validation	Read the testing data set and process it with the adjusted weights of the newly trained model and test the estimated result with the actual board score of testing data set.

The application will read the adjusted weight parameters from theta files and read input parameters from the user which are used and processed and finally gives the predicted score. Firstly, weight is adjusted based on the number of nodes in hidden layer and training datasets. After the weights of the neural network is obtained the network is tested using testing datasets. When testing the datasets the RMSE of the network is obtained. If the RMSE of the dataset is high then the result given by the system will have high chance of giving error. So if the RMSE is high the neural network needs to be trained again to adjust the weights of neural network and to obtain the adjusted weight when RMSE is minimum.

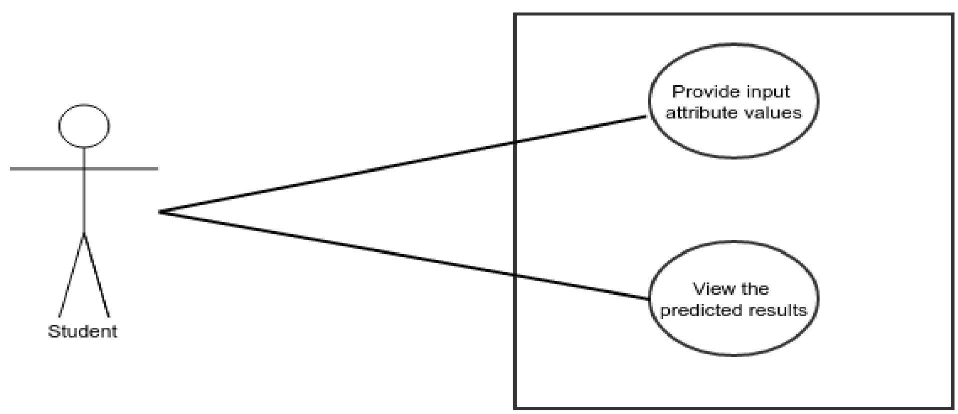


Figure 2 - Use Case diagram of prediction

As illustrated in the Figure 2 above, the student will provide the input parameter values like midterm score, pre-board score, assignment score, internal score, and attendance score the input given will be changed into matrix form and multiplied with the value of theta (which are in matrix form) obtained in the neural network. The result obtained by the multiplication will be the result of the prediction.

2.3 Feasibility Analysis

2.3.1 Schedule feasibility

The time allocated for this system to develop is about four months and several tasks to be performed can be divided to do on weekly basis. Time allocation for different tasks can be tabulated in Table 2 as below:

Table 2 - Tasks Duration with Precedence

Index	Activity	Duration	Precedence
A	Paper reading and title selection	1	-
B	Data Collection and normalizing data	1	A
C	Documentation	8	A
D	Code development for training	5	A, B
E	Testing and validation	2	A,D
F	Code for prediction	3	A,D,E

The total duration needed to complete the task is 20 weeks and with a group of two the above tasks can be completed in time. The time allocation of these tasks can be represented in activity network diagram with precedence as in Figure 3 and Figure 4 below:

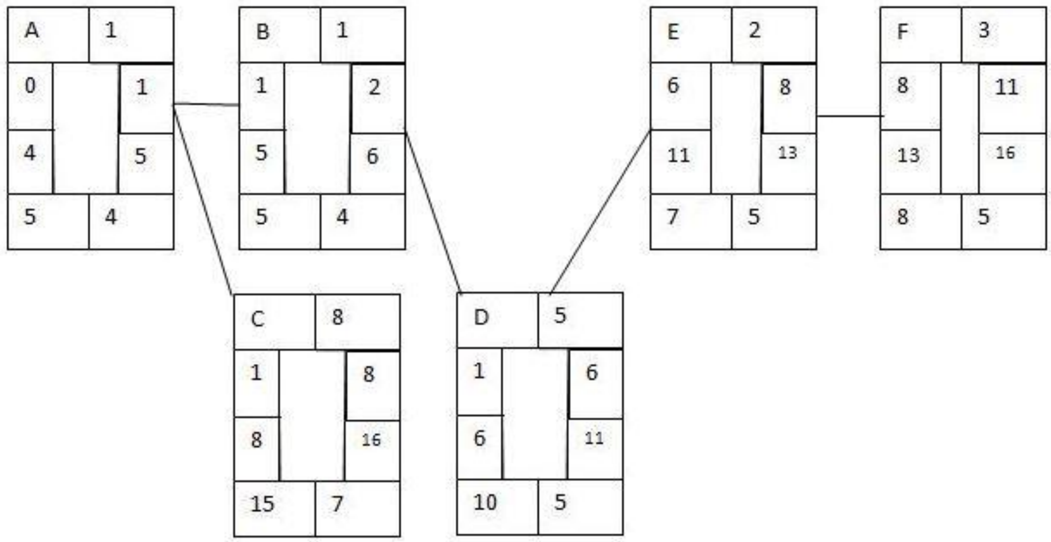


Figure 3 - Activity network diagram

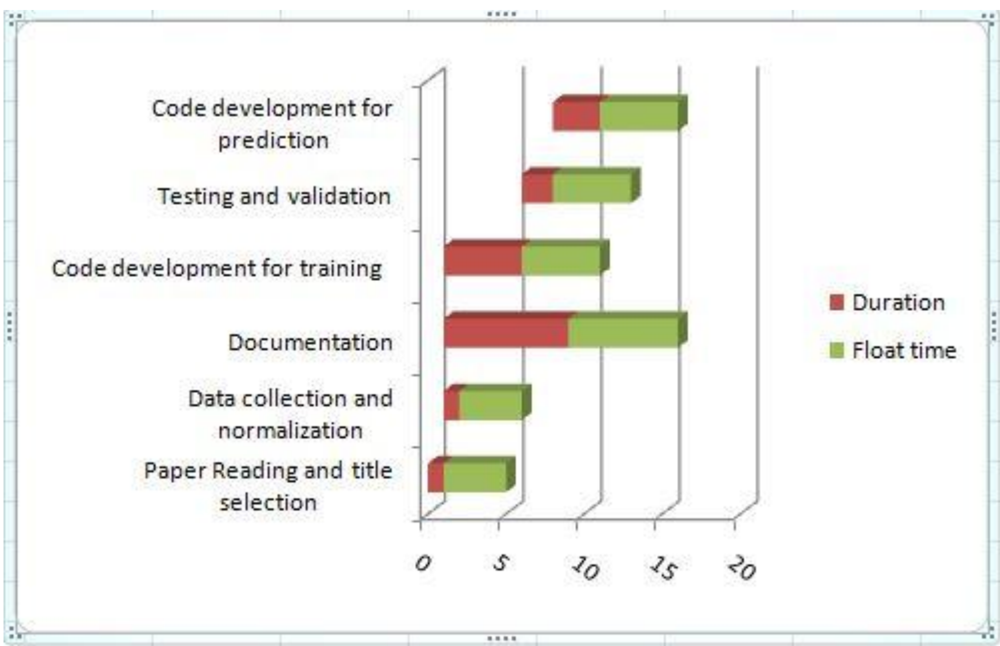


Figure 4 - Gantt chart

2.3.2 Technical feasibility

Student Board Prediction System is a web application that uses Grails Framework. It uses JavaScript for validating the user inputs, Groovy Server Pages (GSP) for front end design and Java as back end. It works as a client server architecture model. This system is platform independent as it runs in browser like Mozilla or Google chrome. For the purpose of training, on open source GNU Octave is used and several API's are available in Octave for different modules needed. As all the technology required to develop this application is available, it is determined to be technically feasible.

2.3.2 Operational feasibility

This system works under two-tier client server architecture model where end user makes a request to get predicted score while the server process the user inputs and respond providing the predicted score. Data can be hosted in a server and client can access it through the Internet. In this way, the system is determined to be operationally feasible.

CHAPTER 3 : SYSTEM DESIGN

3.1 Methodology

Several factors that can affect the performance of the students were considered and studied with greater care. From the several factors, few major factors that has greater impact on performance of students were listed and studied. These influencing factors were listed and considered as an input variables for the neural network topology under study.

We have collected the data from DWIT college and altogether data from 195 students were collected and 20 percent of the data were separated for testing purpose. The training data set were mean normalized. Programming language Octave was used for the purpose of training the data set. Different API's and modules are available in this language and is an open source language. After the collection of data, some of the missing values are replaced by the average value. After normalizing the data, these are separated as training set and testing set. The training data set were feed to the training system develop in Octave language and value of different weights of neural network were obtained as separate theta files. These theta files contains the value of parameter of different adjusted parameter needed for training purpose. For the purpose of testing, these theta parameters were read from the theta files and estimated values were calculated after matrix multiplication of input parameters and theta parameters and finally the output matrix of size one by one was obtained which was the final predicted value. These estimated predicted value was compared with the original board score from testing data for validation purpose. The process adopted in prediction process can be represented as Figure 5 below:

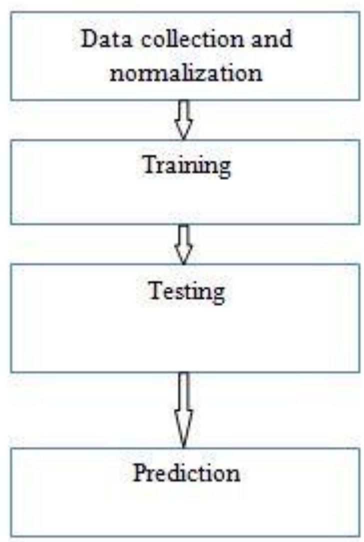


Figure 5 - Steps in prediction

3.1.1 Data collection and normalization

First of all, the data were collected from DWIT college administration. Data of 195 students were collected and listed and these data were analyzed properly. Among the different values, only the needed attributed were listed and other values were filtered during the process. Attributes like midterm score, pre-board score, attendance score, internal score, assignment score, and board score were taken as the parameters. Raw data that were collected are listed in appendices I, II, and III respectively. Sample data after filtering is represented in the Table 3 as shown below:

Table 3 - Sample data collected after filtering

Percentage in mid-term score	Percentage in pre-board score	Assignment score	Internal score	Attendance score in percentage	Percentage in board score
71	56	72	66	69	70
74	61	55	73	79	72
53	59	46	76	68	69
52	47	67	56	59	71

After the process of filtering, these data were normalized obtaining the percentage score in the unit between 0 and 1. The formula that was used for normalizing the data is as below:

normalized percentage value = (obtained value) / 100

The sample data set of normalized value is shown in Table 4 below:

Table 4 - Sample normalized data used for prediction

Percentage in mid-term score	Percentage in pre-board score	Assignment score	Internal score	Attendance score in percentage	Percentage in board score
0.71	0.56	0.72	0.66	0.69	0.70
0.74	0.61	0.55	0.73	0.79	0.72
0.53	0.59	0.46	0.76	0.68	0.69

3.1.2 Training

For the purpose of training, about 80 percent of the data set were separated that includes about 1792 data items. Whole training data were kept in two files one containing the input parameters while the other file containing the board score. All data items used were mean normalized. Sample data set separated for training purpose is represented in Table 5 that contains attributes values as mid-term score, pre-board score, attendance, internal score, and assignment score. Similarly, Table 6 contains board score for training purpose.

Table 5 - Sample data set for training

Mid-term score	Prebard score	Attendance	Internal Score	Assignment
0.52	0.7942	0.88	0.85	0.76105
0.4417	0.5042	0.89	0.75	0.646475
0.4133	0.35	0.84	0.68	0.570825
0.4417	0.6875	0.85	0.81	0.6973
0.6183	0.8208	0.86	0.91	0.802275
0.7283	0.6017	0.82	0.84	0.7475
0.6583	0.6533	0.81	0.86	0.7454
0.4467	0.4783	0.73	0.75	0.60125
0.3717	0.4283	0.77	0.73	0.575
0.325	0.4658	0.79	0.69	0.5677

Table 6 - Sample data set for training with board score

BoardScore
0.726
0.65
0.652
0.752
0.784
0.76
0.696
0.582
0.45
0.604

3.1.3 Testing

For the purpose of testing, about 20 percent of the data set were separated that includes about 49 data items. Whole testing data were kept in two files one containing the input parameters while the other file containing the board score. Sample data set separated for training purpose is represented in Table 7 with attributes variables as mid-term score, pre-board score, attendance, internal score, and assignment score respectively. Similarly, Table 8 contains the target value as board score for validating the system.

Table 7 - Sample data set for testing

Mid-term score	Prebard score	Attendance	Internal Score	Assignment
0.790476191	0.21	0.71	0.86	0.642619
0.525833333	0.385375	0.8	0.570402778	0.570403
0.3764	0.3559	0.83	0.71	0.568075
0.6736	0.619	0.81	0.89	0.74815
0.727	0.770238095	0.78	0.8	0.76931
0.63	0.5985	0.76	0.75	0.684625
0.339791667	0.386375	0.72	0.482055556	0.482056
0.606944444	0.469444444	0.81	0.628796296	0.628796
0.3533	0.5317	0.7	0.74	0.58125
0.6292	0.703303571	0.84	0.8	0.743126

Table 8 - Sample data set for testing with board score

BoardScore
0.596
0.54
0.654
0.7166667
0.772
0.78
0.45
0.53
0.614
0.756

These testing data items were read one by one and used for validation of the system. The adjusted weights from training data set and the test parameters were used to calculate the estimated board score. The estimated board score were finally compared with the actual test board score to calculate the accuracy of the system through manual process.

3.1.4 Prediction

After the training and testing phase is complete, next step is to follow the process of prediction. After the successful training, the weights of neural network are adjusted and taking the input parameters from the user, the system can predict the board score. The ultimate score predicted by the system with user interface is listed in the appendices I, II, and III respectively.

3.1.5 Input variables

The input variables considered as influencing factors selected are those which are easily accessible from administrative department. Lists of input variables are:

1. Mid-term result average score
2. Pre-board result average score
3. Assignments submitted
4. Internal assessment score
5. Attendance

All these factors were converted into a suitable format that can be used for neural network analysis.

3.1.6 Output variable

The output variable is a single variable that gives the predicted board score of any individual students.

3.1.7 Topology of the network

The data were collected from the administrative department and were converted into suitable format for neural network analysis. Multilayer perceptron was chosen among the various network topologies like recurrent network, time-lagged recurrent network. Different network has its own trade off speed for accuracy, performance, and convergence for optimality (Oladokun, 2008).

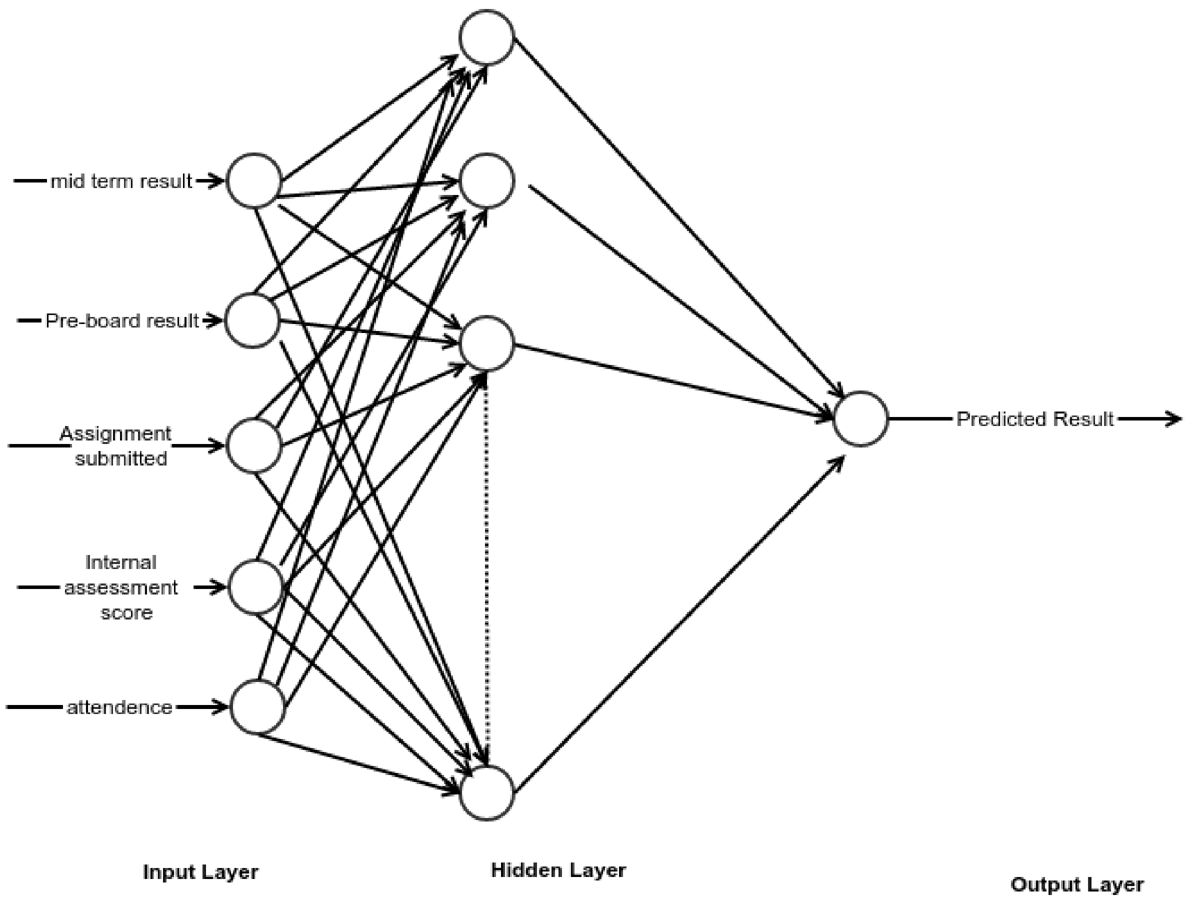


Figure 6 - Neural network topology

Figure 6 represents the MLP which are layered feed forward networks which are basically trained with static back propagation. The main advantage of this network is that it is easy to use and they are capable for easy input/output mapping. The main disadvantage of this network is that it consumes more time for training and requires a lot of training data. Hit and trial method is adopted for determining the number of processing elements and hidden layers required in the network. Selecting large number of hidden layers will slow down the training time while small number of hidden layers may lower the processing capability. Training is performed firstly fixing hidden layer to one and slowly increasing the number of processing elements (nodes) in that layer. Then hidden layer is change to two and effects on cost function while increasing and decreasing the processing elements studied. In this way, neural network topology was set from hit and trail method studying the behavior of the cost function. When cost function reached the minimal, we stop the process and set the network topology for the prediction. Sample table output for hit and trial method for network topology selection is tabulated in Table 9 below:

Table 9 - Cost function obtained after hit and trial in training

Nodes in first layer	Nodes in second layer	Nodes in third layer	Number of Iterations	Cost function value
50	50	50	70	2.466435e+02
10	20	30	90	2.466435e+02
20	20	20	131	2.466436e+02
10	40	80	222	2.466435e+02

Whole data set was divided into training and testing set to adopt supervised learning approach. About 80% of the data were used for training purpose while 20% of the data were used for testing purpose. Because of the limited number of data, after separating the data set for testing purpose, the training data set were replicated to increase the number of data set. A total of 195 data items were collected from the students of different batches for the analysis.

After the data classification, the neural network topology was built based on the multilayer perceptron with three hidden layers and 20 processing elements per layer. During the training process, number of hidden layers and number of nodes in each layer were set by a hit and trial method.

For the feed forward approach in network, vector implementation was proceeded as:

$z(2) = \theta(1)*a(1)$

$a(2) = g(z(2))$

Bias is fixed as $a0(2) = 1$

$z(3) = \theta(2)*a(2)$

$h\theta(x) = a(3) = g(z(3))$

where both $z(2)$ and $a(2)$ are represented in vector notation.

After the training is complete, the estimated board score is compared with the original test score from testing data set to calculate the RMSE in prediction. The formula used to calculate RMSE is given below:

RMSE = square root of $\sum$(estimated score - original score)^2/n where n is the total number of testing data set.

Similarly, the accuracy of the system is determined with a formula as represented below:

System Accuracy = (1 - error) * 100%

where error is the difference between the target score and estimated score obtained during the testing process. Accuracy was determined for each testing data set and the average was calculated to find the system overall accuracy. The final output of the system while testing is listed in appendices II and III.

3.2 Algorithm

Back propagation algorithm is used for training the neural network. Algorithm implemented for back propagation can be illustrated as below:

3.2.1 Steps for back propagation

1. Initialize the weights or parameters to small random values (-0.5 to 0.5).

2. Feed the training sample through the network and determine the final output.

3. Compute the error for each output unit

   for unit k, it is

   $$\delta k = (tk - yk)*f'(yink) = (tk - yk)*f(yink)*[1-(f(yink))]$$

4. Calculate the weight correction term for each output unit

   for unit k, it is

$$\Delta\theta jk = \alpha\delta k\ Zj$$

5. Propagate the delta terms (errors) back through the weights of hidden units where the delta input for jth hidden unit is

$$\delta j = (\delta ink)*f'(Zink) = (\delta ink)*f(Zink)*[1-(f(Zink))]$$

6. Calculate the weight correction term or parameters for the hidden units

$$\Delta\theta ij = \alpha\delta j\ xi$$

7. Update the weights or parameter

$$\theta jk\ (new) = \theta jk\ (old) + \Delta\theta jk$$

8. Test for the stopping (maximum cycles, small changes).

3.3 System Design

3.3.1 Class diagram

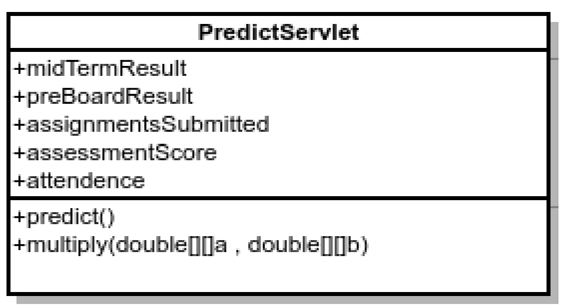


Figure 7 - Class diagram

Figure 7 represents the class diagram where the end user of this system is the student and one of the major class is PredictServlet class with input parameters to system as its instance variables with a major method as predict method and multiply method.

3.3.2 Event diagram

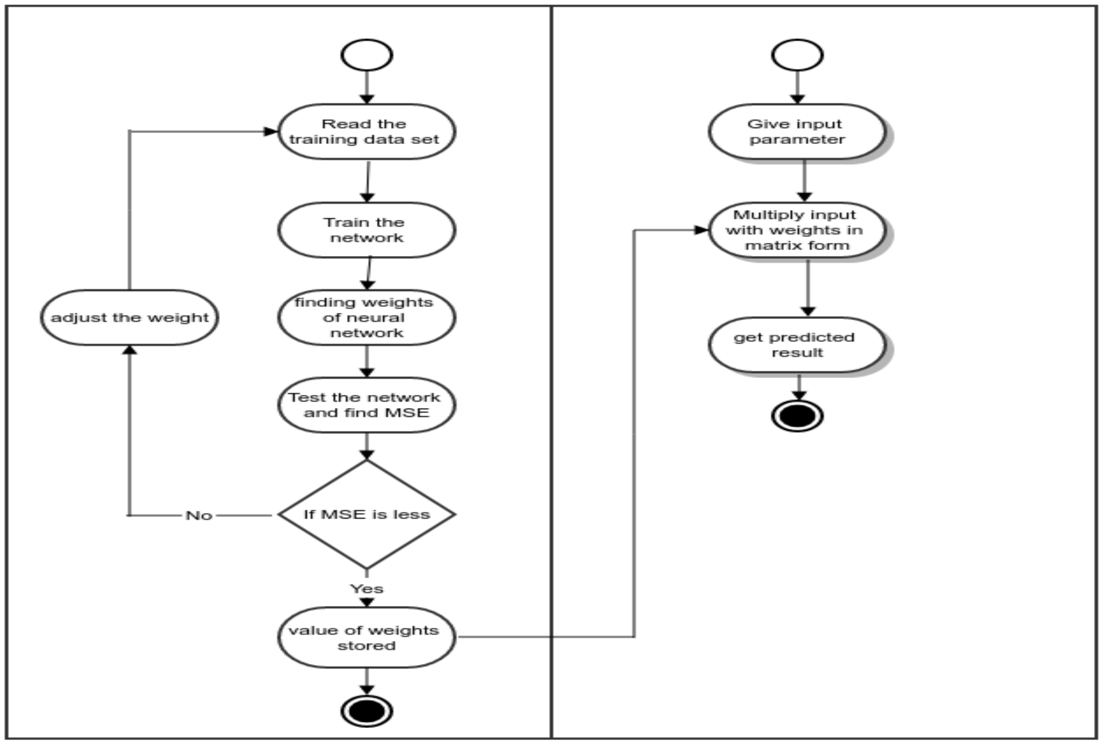


Figure 8 - Event diagram for prediction

Figure 8 illustrates the event diagram for prediction where the developer will develop the system to train with the available dataset. Number of hidden layers and the number of nodes in each hidden layers are chosen such that the cost function derived from training is minimum. After the training is complete, the adjusted weights are used and from testing dataset, it is tested to see if the system can predict accurately with acceptance level of error or threshold. If the training is successful, then user inputs are provided and the system is ready to predict with given input parameters.

3.3.3 Sequence diagram

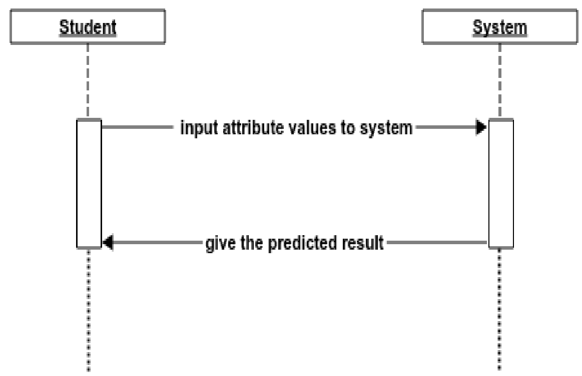


Figure 9 - Sequence diagram for prediction

From Figure 9 which shows that the end user of this system are students from DWIT who will use the system providing the given input parameters, and students are able to get their predicted board score.

CHAPTER 4 : IMPLEMENTATION AND TESTING

4.1 Tools Used

There are several tools and technologies that are used to complete the project. These tools are summarized briefly as following:

4.1.1 Octave

For the purpose of training the data set, an open source Octave is used. There are several modules or functions available for optimizing the cost function during training phase. Different functions and modules are available in Octave to set up input, hidden layers, and generate output. It is easy to interpret the result in this language and is used for training the dataset in the project.

4.1.2 Creately/Gliffy

Creately and Gliffy are used as a CASE tool for making diagram and design software. These are efficient tools in making technical diagrams with simple drag and drop technique. They play effective role in visualization of the project. In the project, different diagram like UML class diagram, event diagram, and sequence diagram are prepared in Creately as well as Gliffy.

4.1.3 HTML/CSS/JSP/JavaScript

HTML , CSS, and JSP are used for the front end design of the application. These are used for making the user interface user friendly. Similarly, JavaScript is used in validating the user inputs.

4.1.4 Java Servlet

Java Servlet is used to built the web-based application. Programming language Java is used to develop the application for prediction. The adjusted weights are saved in different theta files with Octave during training and these files are read through Java language and reading the user input and these theta files in java, it finally gives the predicted score.

4.1.5 IDEA Intellij

IDEA Intellij is a Java integrated development environment (IDE) tools that is used for developing computer software. The project is written, compiled and run in this software.

4.2 Listing of Major Classes and Modules

4.2.1 PredictServlet class

It is the main class that is included in the Servlet program is PredictServlet. Major instance variables and methods of this class is represented in code as below:

```
@WebServlet(name = "PredictServlet")

public class PredictServlet extends HttpServlet {
```

```
//class that returns value after multiplication

public static double[][] multiply(double[][] a, double[][] b) {

int rowsInA = a.length;

int columnsInA = a[0].length; // same as rows in B

int columnsInB = b[0].length;

double[][] c = new double[rowsInA][columnsInB];

for (int i = 0; i < rowsInA; i++) {

  for (int j = 0; j < columnsInB; j++) {

     for (int k = 0; k < columnsInA; k++) {

        c[i][j] = c[i][j] + a[i][k] * b[k][j];

        }

     }

  }

 return c;

 }
```

This class consists of five input variables as instance variables with two major methods predict() and multiply(). The method multiply() is used for the two dimensional matrix multiplication where it takes two matrices as argument and after multiplication, it returns two dimensional matrix. Similarly, when predict() method is called, it will return the predicted board score after initializing the user input parameters in its class instance variables and processing these with values of reading adjusted theta files.

4.3 Testing

For the purpose of system validation, the model developed with adjusted weights after training of neural network was tested with inputs from testing data set. For each set of testing data set as input, an output score was noted. Similarly, the difference between the original board score from testing data set and estimated predicted value derived while testing was studied. The difference in the value of original board score and estimated board score derived after testing is represented as given below in Table 10.

Table 10 - Comparison between target and estimated score for system validation

Mid term	Preboard	Attendence	Internal score	Assignment	Target Value (Y)	Estimated predicted score(Y')	Y - Y'
0.790476	0.21	0.71	0.86	0.642619048	0.6	0.5628	-0.0372
0.525833	0.385375	0.8	0.570402778	0.570402778	0.54	0.55807	0.01807
0.3764	0.3559	0.83	0.71	0.568075	0.57	0.52096	-0.04904
0.6736	0.619	0.81	0.89	0.74815	0.72	0.68785	-0.03215
0.727	0.770238	0.78	0.8	0.769309524	0.77	0.73872	-0.03128
0.63	0.5985	0.76	0.75	0.684625	0.78	0.66572	-0.11428
0.339792	0.386375	0.72	0.482055556	0.482055556	0.45	0.51007	0.06007
0.606944	0.469444	0.81	0.628796296	0.628796296	0.53	0.61064	0.08064
0.3533	0.5317	0.7	0.74	0.58125	0.61	0.58289	-0.02711
0.6292	0.703304	0.84	0.8	0.743125893	0.76	0.70311	-0.05689
0.588889	0.488889	0.82	0.63	0.631944444	0.63	0.61421	-0.01579
0.236667	0.126563	0.85	0.404409722	0.404409722	0.38	0.38285	0.00285
0.758333	0.659722	0.85	0.78	0.762013889	0.76	0.71184	-0.04816
0.629579	0.514833	0.81	0.14	0.523603125	0.7	0.60758	-0.09242
0.5	0.3526	0.87	0.72	0.61065	0.61	0.54891	-0.06109
0.7403	0.572	0.79	0.85	0.738075	0.67	0.68295	0.01295
0.450417	0.477604	0.83	0.586006944	0.586006944	0.59	0.57821	-0.01179
0.303125	0.397313	0.79	0.4968125	0.4968125	0.41	0.50829	0.09829
0.705556	0.378472	0.77	0.618009259	0.618009259	0.5	0.59662	0.09662

Above Table 10 represents the inputs from training data set and for each data set of training, the respective estimated score was calculated from the model of neural network with adjusted weights from training. These test data set are not used for training purpose and are only used after the training is complete. "Y" represents target value of original board score while "Y' " represents estimated score. The difference between these values are represented in column Y - Y' column which is the error in prediction. In total 49 data set were used for testing purpose and above values were used to calculate the value of RMSE. Overall RMSE was determined to be 0.020488354 from the formula listed in above section 3.1.7.

CHAPTER 5 : MAINTENANCE AND SUPPORT

The academic performance of the student in any college may change according to time. Any particular batch of student will study for four years to complete Bachelor level in CSIT. Every year new students are enrolled in the college. Hence, to bring consistency in prediction in system, the training of data set at frequent interval of time is needed. Training the data set at frequent interval from four to five years can be one of the maintenance measure for accuracy of the system. Similarly, environmental and psychological factors of the students also plays vital role in academic performance of the students. So, consideration of these factors in prediction can be another maintenance and support measure for efficient prediction of academic performance.

CHAPTER 6 : CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The project Student Board Score Prediction based on neural network was completed successfully. The project is able to predict the final score of the students. The system was calibrated on the basis of attributes like mid-term score, pre-board score, attendance, internal score, and assignment score. The validation of the system was carried out using RMSE which accounts 95 percent accuracy.

6.2 Recommendation

Attribute selection is one of the key factor that determines the accuracy of prediction. So, considering psychological and environmental factors can improve the performance in prediction of students academic performance. Adoption of these factors along with academic factors can help in improving the performance of the system in prediction.

APPENDIX I

1. Score on particular subject from one batch

Subject_1	F.M.1	P.M.1	O.M.1	Percentage_1	Status_1
1.CSC-251: Theory of Computation	80	32	58	72.50	PASS
1.CSC-251: Theory of Computation	80	32	13	16.25	FAIL
1.CSC-251: Theory of Computation	80	32	47	58.75	PASS
1.CSC-251: Theory of Computation	80	32	60	75.00	PASS
1.CSC-251: Theory of Computation	80	32	68	85.00	PASS
1.CSC-251: Theory of Computation	80	32	49	61.25	PASS
1.CSC-251: Theory of Computation	80	32	64	80.00	PASS
1.CSC-251: Theory of Computation	80	32	44	55.00	PASS
1.CSC-251: Theory of Computation	80	32	44	55.00	PASS
1.CSC-251: Theory of Computation	80	32	19	23.75	FAIL
1.CSC-251: Theory of Computation	80	32	40	50.00	PASS
1.CSC-251: Theory of Computation	80	32	44	55.00	PASS
1.CSC-251: Theory of Computation	80	32	56	70.00	PASS
1.CSC-251: Theory of Computation	80	32	51	63.75	PASS
1.CSC-251: Theory of Computation	80	32	34	42.50	PASS
1.CSC-251: Theory of Computation	80	32	34	42.50	PASS
1.CSC-251: Theory of Computation	80	32	51	63.75	PASS
1.CSC-251: Theory of Computation	80	32	52	65.00	PASS
1.CSC-251: Theory of Computation	80	32	39	48.75	PASS
1.CSC-251: Theory of Computation	80	32	46	57.50	PASS
1.CSC-251: Theory of Computation	80	32	32	40.00	PASS

2. Score on particular subject from one batch

Subject_2	F.M.2	P.M.2	O.M.2	Percentage_2	Status_2
CSC-252: System Analysis and Design	60	24	49	81.67	PASS
CSC-252: System Analysis and Design	60	24	26	43.33	PASS
CSC-252: System Analysis and Design	60	24	52	86.67	PASS
CSC-252: System Analysis and Design	60	24	56	93.33	PASS
CSC-252: System Analysis and Design	60	24	56	93.33	PASS
CSC-252: System Analysis and Design	60	24	53	88.33	PASS
CSC-252: System Analysis and Design	60	24	50	83.33	PASS
CSC-252: System Analysis and Design	60	24	42	70.00	PASS
CSC-252: System Analysis and Design	60	24	48	80.00	PASS
CSC-252: System Analysis and Design	60	24	35	58.33	PASS
CSC-252: System Analysis and Design	60	24	37	61.67	PASS
CSC-252: System Analysis and Design	60	24	42	70.00	PASS
CSC-252: System Analysis and Design	60	24	54	90.00	PASS
CSC-252: System Analysis and Design	60	24	45	75.00	PASS
CSC-252: System Analysis and Design	60	24	43	71.67	PASS
CSC-252: System Analysis and Design	60	24	39	65.00	PASS
CSC-252: System Analysis and Design	60	24	46	76.67	PASS
CSC-252: System Analysis and Design	60	24	43	71.67	PASS
CSC-252: System Analysis and Design	60	24	36	60.00	PASS
CSC-252: System Analysis and Design	60	24	44	73.33	PASS
CSC-252: System Analysis and Design	60	24	32	53.33	PASS

3. Attendance and assignment score from any one batch

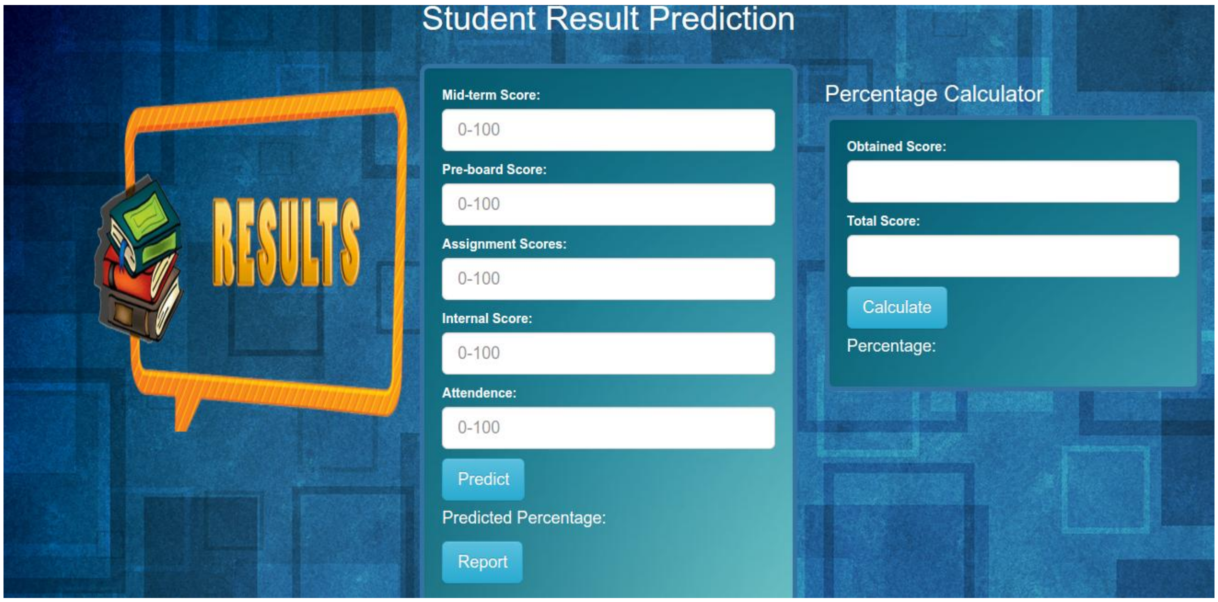
Total Assignment_6	Submited Assignment_6	Total Attendance_6	Actual Attendance_6	Attendance Percentage_6	Overall Percentage
3	3	22	22	100.00	74.44
3	3	22	20	90.91	40.00
3	3	22	22	100.00	77.50
3	3	22	19	86.36	80.56
3	3	22	22	100.00	86.39
3	3	22	21	95.45	75.56
3	2	22	22	100.00	76.39
3	3	22	20	90.91	62.50
3	3	22	21	95.45	63.61
3	2	22	20	90.91	52.78
3	3	22	22	100.00	56.11
3	3	22	20	90.91	63.89
3	3	22	18	81.82	80.83
3	3	22	20	90.91	71.39
3	3	22	22	100.00	55.28
3	3	22	22	100.00	58.06
3	2	22	22	100.00	61.39
3	3	22	21	95.45	72.22
3	2	22	22	100.00	63.89
3	3	22	22	100.00	72.78

4. Midterm, final term, and board score from any one batch

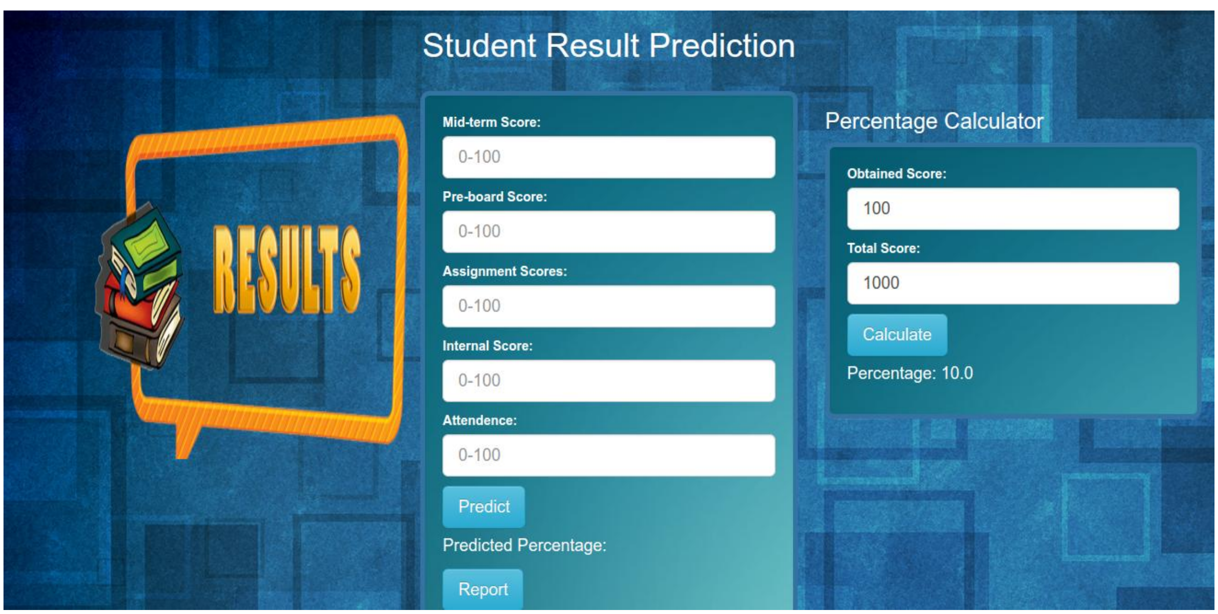
SemesterNo	MidTermPercentage	FinalPercentage	BoardPercentage
1	52	79.42	72.6
1	44.17	50.42	65
1	41.33	35	65.2
1	44.17	68.75	75.2
1	61.83	82.08	78.4
1	72.83	60.17	76
1	65.83	65.33	69.6
1	40	23.67	57.6
1	44.67	47.83	58.2

APPENDIX II

1. Home Page of application



2. Validation and percentage calculation



3. Background process with report about input matrices and parameters values

Report of Result Prediction

Back

Input Matrix	Theta1	Theta2	Result after first multiplication	Result after second multiplication	Predicted Score
	0.09971497215807268 -0.005700795412143862 0.04353486617492459 0.0272449541720675 -0.005369718904100168 -0.04179790578757754 -0.05245744503685398 0.08454235772700212 0.03671332792125229 0.02939945700213674 0.03189600862792945 0.02629632509373077 -0.0809207552811496 0.06771300535037987 0.04814984878531905 -0.015683545062983 0.06779186330892512 -0.07330056151306513 -0.0219680149191887 0.0821430882270665 0.01181576615189881 0.119493825075277 -0.06459559706026599 -0.09524794609074139 -0.08447386377954674 -0.090442981640387 0.005686258471964747 -0.06196114740916857 0.06060305370922259 -0.1156361035028855 0.1104469425937995 -0.05986565384688041 0.09086081492676076 -0.06511869712850764 -0.03495655335073026 0.1091789396625728 -0.09698356180406659 -0.001939491618365305 -0.1192546617496354 0.09685975623422441 -0.125683403660551 0.06159491677800091 0.1262746065698356 -0.0250207353081985 -0.07192833769257088 -0.04731546056081709 0.0853583088147989 0.1116156486175692 -0.01907631053288877 -0.01264591856969333 , -0.2555107483169635 -0.03731945533873332 -0.04857064586424635 0.3097354890771459 -0.2417976410574103 0.1584217399999254 0.2793381814624837 -0.06260481702133901 -0.03328599464946686 -0.00969414538418057 -0.0811048760926949 -0.2497528637302764 0.2203660455117437 -0.1191413935106142 0.3555366961583356 0.3511425855879701 0.1184930650704703 0.09792134323961205 -0.1620955156666752 0.1674874296366446 -0.02974866949578127 -0.06853630781836356 0.05534359155228086 0.08404735979289331 0.02560956668919494 0.284670147872068 -0.2098106384336453 -0.2602957012171928 -0.04745945859118435 0.1537940912880528 -0.02134760768607882 -0.1383448458984793 -0.2884095623405151 -0.3753860331189821 -0.1210094179748616 0.1122609040299347 0.01399754026577077 0.03636317393719277 0.1380507960931361 -0.167796275467868 0.3848509144309266 0.04445040042900531 -0.1512617859777383 0.1799954005280902 -0.02004770680951737 0.1727024710778871 -0.008546782826833183 0.0917991186722266 0.1701191038371926 0.2660627854505892 , -0.5757034399676633 0.02672288771348475 -0.1643983253511451 0.2355211927393603 -0.4365595447820167 0.1690340590778925 0.300843746563227 -0.2301921353691227 0.05293704232438834 0.02013415928284738 -0.02746318942615242 -0.5004975860449785 0.3528097216096665 -0.008616312854597418 0.6399994014923732 0.3941043794720198	-0.1014544099165798 , -0.7244060387302553 , -0.05405273146495643 , -0.2565488825641627 , 0.3107119354513955 , -0.6025634429829974 , 0.1602126854302612 , 0.3323645568118568 , -0.2204932752649711 , -0.02319647409828739 , -0.01856686314026908 , -0.1488099119555911 , -0.7552186734204286 , 0.3073844076668502 , -0.1000237900021806 , 0.6629045800617978 ,	1.0 0.3894177978119276 0.4798752011791753 0.5040765311662858 0.6206641752585282 0.38074942857584976 0.5403952162358961 0.6139800368913599 0.4668121194176398 0.5371048678643003 0.49961988565244664 0.44730043925633356 0.3862723812902041 0.5828625629206461 0.5233633125747905 0.6498285963181671 0.6370657618870581		

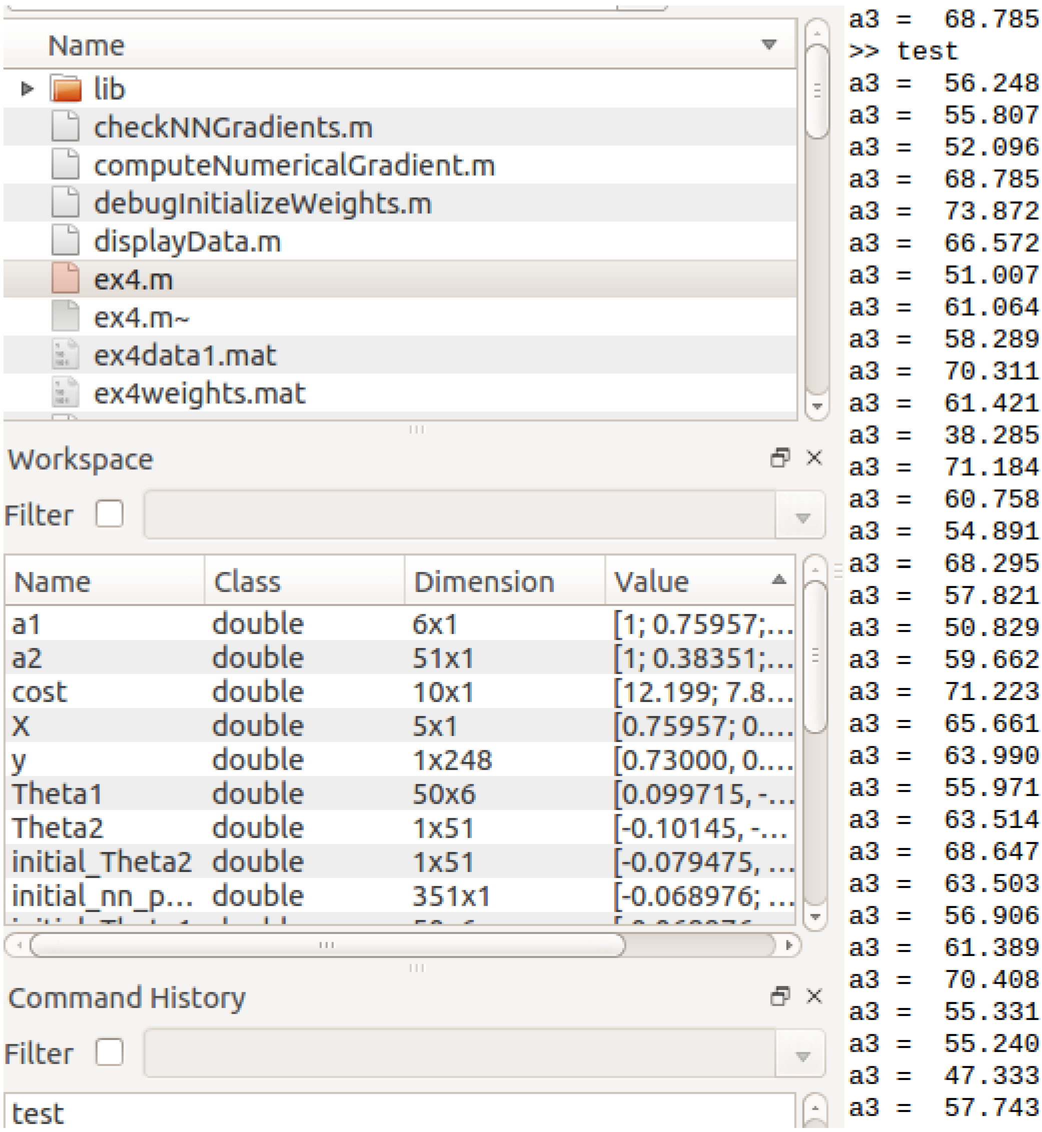
APPENDIX III

1. Inputs of testing data set for system validation

test.m

```
Theta1 = load('Theta1').Theta1;
Theta2 = load('Theta2').Theta2;
#X = [0.5 0.5 0.5 0.5 0.5]';

X = [0.7904761905 0.21 0.71 0.86 0.6426190476]';
a1 = [ones(size(X(1, :))); X];
a2 = sigmoid(Theta1 * a1)';
a2 = [ones(size(a2(:, 1)), 1) a2]';
a3 = sigmoid(Theta2 * a2) * 100


X = [0.5258333333 0.385375 0.8 0.5704027778 0.5704027778]';
a1 = [ones(size(X(1, :))); X];
a2 = sigmoid(Theta1 * a1)';
a2 = [ones(size(a2(:, 1)), 1) a2]';
a3 = sigmoid(Theta2 * a2) * 100



X = [0.3764 0.3559 0.83 0.71 0.568075]';
a1 = [ones(size(X(1, :))); X];
a2 = sigmoid(Theta1 * a1)';
a2 = [ones(size(a2(:, 1)), 1) a2]';
a3 = sigmoid(Theta2 * a2) * 100

X = [0.6736     0.619    0.81    0.89    0.74815]';
a1 = [ones(size(X(1, :))); X];
a2 = sigmoid(Theta1 * a1)';
a2 = [ones(size(a2(:, 1)), 1) a2]';
a3 = sigmoid(Theta2 * a2) * 100

X = [0.727      0.7702380952    0.78    0.8     0.7693095238]';
a1 = [ones(size(X(1, :))); X];
a2 = sigmoid(Theta1 * a1)';
a2 = [ones(size(a2(:, 1)), 1) a2]';
a3 = sigmoid(Theta2 * a2) * 100
```

2. Output of estimated score of the testing data set



REFERENCES

Bahadır, E. (2016). Using Neural Network and Logistic Regression Analysis to Predict Prospective Mathematics Teachers' Academic Success upon Entering Graduate Education.

Bogdan Oancea, R. D. (2016). Predicting students' results in higher education using neural networks.

Kabakchieva, D. (2012). Student Performance Prediction by Using Data Mining Classification Algorithms .

Oladokun, V. A. (2008). Predicting Students' Academic Performance using Artificial Neural Network.

Syed Tahir Hijazi, S. R. (2014). Factors affecting students' performance.