AN ENHANCED FEED-FORWARD NEURAL NETWORKS AND A RULE-BASED ALGORITHM FOR PREDICTIVE MODELLING OF STUDENTS' ACADEMIC PERFORMANCE

AJIBOYE ADELEKE RAHEEM

Doctor of Philosophy (Computer Science)

UNIVERSITI MALAYSIA PAHANG

THESIS CONFIDENTIAL STATUS
UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THE THESIS AND COPYRIGHT

Author's full name : AJIBOYE ADELEKE RAHEEM

Date of birth : SEPTEMBER 16, 1973.

Title AN ENHANCED FEED-FORWARD NEURAL NETWORKS AND A RULE-BASED ALGORITHM FOR PREDICTIVE MODELLING OF STUDENTS' ACADEMIC PERFORMANCE.

Academic Session : 2015/2016

I declared that this thesis is classified as:

☐ **CONFIDENTIAL** (Contains confidential information under the official Secret Act 1972)

☐ **RESTRICTED** (Contains restricted information as specified by the organization where research is done)

☑ **OPEN ACCESS** I agree that my thesis to be published online as open access (Full text)

I acknowledge that Universiti Malaysia Pahang reserves the right as follows:

1. The thesis is the property of Universiti Malaysia Pahang.
2. The library of Universiti Malaysia Pahang has the right to make copies for the purpose of the research only.
3. The library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature) Signature of Supervisor)

A 05770540
Passport Number
Date:

Name of Supervisor
Date:

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy in Computer Science.

____
(Supervisor's Signature)

FULL NAME : Ruzaini Abdullah Arshah

POSITION :

DATE :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citation which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Author's Signature)

FULL NAME : Ajiboye Adeleke Raheem

ID NUMBER : PCC13002

DATE :

AN ENHANCED FEED-FORWARD NEURAL NETWORKS AND A RULE-BASED ALGORITHM FOR PREDICTIVE MODELLING OF STUDENTS' ACADEMIC PERFORMANCE

AJIBOYE ADELEKE RAHEEM

Thesis submitted in fulfilment of the requirements
for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computer Systems & Software Engineering
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2016

Dedicated to my late parents: MR AND MRS HAMZAT AJIBOYE

ACKNOWLEDGEMENTS

My profound gratitude goes to Allah (SWT), the Lord of the world for giving me the opportunity to go this far in my education, may His peace be upon the holy prophet Muhammad (SAW) and all his followers.

I am grateful to the University Malaysia Pahang for the research funding and other opportunities I received from the institution in the course of this research. The knowledge I acquired during the period of this research cannot be quantified. I seek from Allah to give me long life, so that, I can make good use of the quality training impacted to me by this great institution.

I sincerely thank my advisor, Associate Professor Dr Ruzaini Abdullah Arshah, for always there for me. His wealth of experience really counts on this research, and he always welcomes me with or without giving him any prior notice. I also wish to appreciate some useful comments made by Dr Hongwu Qin at the early stage of this research.

I would like to thank my alma mater, University of Ilorin, Nigeria for making my dream a reality. In particular, I am grateful to the former chief executive officer of the institution, Prof. Is-haq Olanrewaju Oloyede; he has touched my life positively in so many ways, I say *JazakAllah Khair*.

I thank my wife, Aminat Titilayo, for her love, understanding and patience at all times, most especially during this programme. I also appreciate the sacrifices made by my children: Abdulbasit, Raheemat, Salim and Azimat. It shall be well with you all.

I thank my late parents for their struggle for me and for always in my support at ensuring that I achieve my heart desires. May Allah be pleased with both of you, overlook your shortcomings and make Al-Janatul Firdaus your final abode. I am grateful to my brothers and sisters, for the show of love to my family while I was on this programme, may Allah reward you all abundantly.

I would also like to thank the staff in the following offices at UMP: all the staff of the Faculty of Computer Systems & Software Engineering, the staff in the Institute of Postgraduate Studies and the staff in the International Office for their support and cooperation at all times.

Again, I thank Allah for giving me the peace of mind and sound health to successfully complete this programme. Finally, I wish to express my gratitude to all my friends and colleagues: Kayode Adewole, Sikiru Abdulsalam, Edward Akindoyo, Julius Odili, Shahid Anwar, Jamila Abdulhadi, Hauwau Kebbe, Hasneeza Liza and everyone who in one way or the other has contributed to the completion of this research. Only Allah can reciprocate the good gesture you have all shown to me.

ABSTRACT

Feed-forward Neural Networks, is a multilayer perceptron and a network structure capable of modelling the class prediction as a nonlinear combination of the inputs. The network has proven its suitability in solving several complex tasks. But sometimes, it has challenges of over-fitting, especially when fitting models from massive data of varied data points. This necessitates its enhancement in order to strengthen its performance. Such enhancement would ensure a predictive network model that can generalize well with a set of untrained data. In this research, in order to alleviate the possibility of over-fitting in a network predictive model, a dynamic partitioning of the dataset is proposed. Also, for a more efficient exploration of students' data collected for this research, a Rule-Based Algorithm is proposed and implemented. The predictive models emanated from the two approaches were evaluated in order to validate their effectiveness. The enhancement done to the Feed-forward Neural Networks (FNN) in the first approach, ensure partitioning of the dataset that is based on the size of the data available for creating the model. The evaluation carried out on the Enhanced Feed-forward Neural Network (EFNN) models show that, there is a decrease in error from 0.261 to 0.029. Similarly, another set of 2000 students' data is trained, the error recorded when the network model is simulated with untrained 500 data show that, there is a reduction in error from 0.0095 to 0.00033. Most of the training performance generated from the network models created also shows that, the EFNN has lower errors and converge faster. The implementation of the rule-based algorithm proposed in the second approach, shows outputs that are consistently accurate. Its efficiency is compared to some existing techniques reported in the literature for the predictive modelling of students' academic performance. Findings from the comparison show that, the proposed RBA explores students' data much better. It can also serve as an alternative algorithm to the use of machine learning techniques in the exploration of students' data for prediction purposes.

ABSTRAK

Rangkaian *Neural Feed-Forward*, adalah *perceptron* pelbagai-lapis dan struktur rangkaian yang mampu memodelkan ramalan kelas sebagai gabungan input-input tidak linear. Rangkaian ini telah terbukti kesesuaiannya dalam menyelesaikan beberapa tugas yang kompleks. Tetapi kadangkala, ia mempunyai cabaran *over-fitting*, terutamanya apabila *fitting* model-model data besar yang mempunyai pelbagai titik data. Ini memerlukan peningkatan dalam usaha untuk mengukuhkan prestasi. Peningkatan itu akan memastikan model rangkaian ramalan yang boleh digeneralisasikan dengan baik melalui satu set data tidak terlatih. Dalam kajian ini, bagi mengurangkan kemungkinan berlakunya *over-fitting* di dalam model ramalan rangkaian, pembahagian dinamik set data dicadangkan. Juga, untuk penerokaan data pelajar yang efisyen, *Rule-Based Algorithm* adalah dicadangkan dan dilaksanakan. Model-model ramalan yang berpunca daripada dua pendekatan telah diuji untuk mengesahkan keberkesanannya. Penambahbaikkan dilakukan kepada *Feed-Forward Neural Network (FNN)* dalam pendekatan pertama bagi memastikan pembahagian set data adalah berasaskan saiz data yang sedia untuk pemodelan. Penilaian yang telah dijalankan ke atas model-model *Enhanced Feed-Forward Neural Network (EFNN)* menunjukkan bahawa, terdapat pengurangan ralat daripada 0.261 kepada 0.029. Satu set 2000 data pelajar lain dilatih dengan cara yang sama, ralat yang direkodkan apabila model rangkaian disimulasikan dengan 500 data tidak terlatih menunjukkan pengurangan ralat dari 0.0095 ke 0.00033. Kebanyakan prestasi latihan yang dijana daripada model-model rangkaian juga menunjukkan EFNN mempunyai ralat rendah dan menumpu dengan cepat. Pelaksanaan algoritma berasaskan peraturan yang dicadangkan dalam pendekatan kedua, menunjukkan output yang konsisten tepat. Kecekapannya dibandingkan dengan beberapa kaedah yang dilaporkan di dalam hasil kajian lepas untuk model ramalan prestasi akademik pelajar. Hasil perbandingan menunjukkan bahawa, RBA yang dicadangkan meneroka data pelajar dengan lebih baik. Ia juga boleh digunakan sebagai algoritma alternatif untuk teknik pembelajaran mesin dalam meneroka data pelajar untuk tujuan ramalan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

φ	Activation function
a_1	Actual value
θ	Bias
η	Learning rate
α	Momentum parameter
ρ_1	Predicted value
ξ_i	Random noise
Q	Size of observations
Σ	Summation

LIST OF ABBREVIATIONS

ANNs	Artificial Neural Networks
DB	Database
CART	Classification and Regression Tree
DM	Data Mining
EDM	Education Data Mining
EFNN	Enhanced Feed-forward Neural Network
ELM	Extreme Learning Machine
FIS	Fuzzy Inference System
FNN	Feed-forward Neural Network
FP	False Positive
GPA	Grade Point Average
HSR	High School Results
IST	Institution's Screening Test
MAE	Mean Absolute Error
MSE	Mean Square Error
NHSEA	Number of High School Examination Attempted
RBA	Rule-Based Algorithm
SA	Student's Age
SNE	Score in National Examination
SO	School Ownership
SSCE	Senior Secondary Certificate Examination
TP	True Positive
UML	Unified Modelling Language

CHAPTER 1

INTRODUCTION

1.1 Background

Vast amount of data are being captured and stored regularly in the institutions of learning, but sometimes, construction of predictive model from this massively stored data does not give the desired outputs in term of accuracy. This may be as a result of the model having the problem of over-fitting. A mechanism should, therefore, be put in place to strengthen the technique used in modelling, to pave way for the building of an accurate predictive model from students' data. This task is vital as information is inevitable for efficient planning and there is need to process data in order to have information.

Predictive modelling of students' data is a key task in Educational Data Mining (EDM). The EDM is a relatively new field of scientific study. Although, for quite a long time, researchers have been capturing and analysing various data that emanate from educational domain. However, only recently has EDM been established as a field in its own right (Scheuer & McLaren, 2012). The EDM is concerned with the development of models to reveal the uniqueness inherent in the data from the educational domain.

The process of EDM transforms the raw data within the education system into valuable information that could make a great potential impact on research and practice (Romero & Ventura, 2010). The focus of this research is on enhancement of the Feed-forward Neural Networks (FNN) technique and the development of a rule-based algorithm that is capable of unveiling useful knowledge from students' data. The word knowledge that is commonly used in data mining has been clarified in the literature. It is not really the kind of knowledge that we carry in our heads, but just a choice of word to refer to the structures that learning methods produce (Witten et al., 2011).

What informed the decision to explore educational data was as a result of the poor academic performance of several newly enrolled undergraduate students in recent times as observed in Nigerian Universites. The review of the literature also shows that, the rate at which students are withdrawn from the university is higher at the early stage of their study. This set of newly enrolled students do not have any continuous assessment or past semester results that can be modelled to predict their future performance. The present research therefore, explored their past academic and demographic achievements.

Identification of predictors of relevant influence has been through the domain experts and findings from the literature. These data were modelled in conformity to the data mining process. Some rationales for seeking information about students at the early stage of their study, is to enable the teachers adopt a teaching methodology that can be of immense benefit to all students irrespective of their learning challenges. It would also help to identify student whose performance is outstanding but in serious need of support. Also, those found to be academically weak could be rightly guided by the counselling unit of the institution by providing them with further assistance that can be tailored directly to their needs.

A good understanding of the students' pattern of enrollment, retention and the rate of dropouts are the key research focus of EDM. The capacity of institutional administrators to clarifies the reason for students drop out, what is responsible for some students that over stay in their study, and so on, is less critical than the ability to accurately predict an instance of such events. The EDM is an emerging interdisciplinary research area, that mainly has to so do with the development of techniques that is suitable for the exploration of dataset that originates from an educational context (Romero & Ventura, 2010).

Exploration of educational data requires building a useful descriptive or predictive model that can be used to describe or predict the hidden information that resides in the data. However, the building of the predictive model from the students' data is the concern of this research. In addition to the use of an enhanced FNN for this task, further focus was to develop a rule-based algorithm that can achieve the resulting model of high accuracy and capable of generalizing well.

Although, with machine learning techniques, achieving a total accuracy is not usually feasible and according to (Negnevitsky, 2011), the issue of accuracy using machine learning such as neural networks is sometime affected by a number of factors. At times, setting of the parameters may be responsible for inconsistent outputs.

One of the main objectives of educational institutions has been identified as to provide high quality education to students (Sundar, 2013). In view of this, a huge budget is usually set aside in most countries to achieve this goal; as the rate of growth of a country can be measured by the quality of its education system (Mishra et al., 2014). In order to achieve the highest level of quality in the institutions of learning, there is a need to discover knowledge from the students' data.

Several ways have been identified by which the quality of education can be improved, the study proposed in (Baradwaj & Pal, 2011) identified alienation of the traditional classroom teaching model, detection of ambiguities in the result sheets of the students and prediction of students' performance at the appropriate time. The embedded knowledge in the educational data set are extractable, this can be achieved by systematic exploration through the data mining techniques.

Data mining is one of the important research fields in computer science and it has been broadly defined as the process of discovering novel and interesting patterns in large amount of data (Jukic et al., 2014). It can also be seen as an area of scientific inquiry that revolves around the development of a series of techniques for making discoveries from educational settings and using those techniques to better understand the students (Baker, 2010).

There are a number of well reported methods used in data mining for the exploration of educational data; prominent among them are machine learning and soft computing methods. Also, data mining has incorporated many techniques from other domains such as database, statistics, pattern recognition, algorithm and visualization (Kantardzic, 2011), to address the extraction of information from large set of data.

The most imperative thing in data mining is model creation of high accuracy that can unveil useful information from data through prediction (predictive mining) or description (descriptive mining). The predictive and descriptive tasks has been identified in (Kumar, 2014) as the two major tasks performed in data mining. The focus of this research is on predictive mining. The approaches proposed in this research addresses the two main tasks in predictive mining i.e classification and regression. While the enhanced feed-forward neural networks predict the expected output (regression), the proposed rule-based algorithm predict the students' performance according to certain labels (classification).

The predictive model is the task of building a model of the target variable as a function of the explanatory variables. A model is a simple representation of a more complicated system and could be static or dynamic (Wu & Coggeshall, 2012). The technique used to build a model from massive data determines the reliability of such model. Also, the relevance of the data been explored is also of utmost importance. This research explored data that comprised of both academic and demographics captured in respect of each student during the admission process. The data is detail and appears to be useful in this instance, as the focus is on the newly enrolled students.

The approaches proposed in this research did not limit the predictors to the certificate qualifications alone, as other relevant attributes considered were found to have added value. Studies reported in the literature has shown that, there is a connection between the students' background and their final grade (Abdous et al., 2012). This implies that, the students' academic background can serve as a useful pointer to the future performance of the students. Therefore, techniques that can model the students' academic antecedents have been chosen in this research.

An enhanced feed-forward neural network trained with Levenberg Marquardt, back propagation algorithm was used for building a predictive model from students' data. The decision to train the network with this algorithm was due to its flexibility, it has shown to be mathematically tractable and most times, it converges to a better solution. Subsequently, in order to achieve a predictive model that efficiently explore students' data, a rule-based algorithm is proposed and was found to be efficient and perform much better than the learning algorithm. The two approaches proposed in this research were found to be suitable for predictive modelling of students' data.

1.2 Problem Statements

Rapid advances in the way data are being captured and stored have enabled organizations to accumulate vast amounts of data; institutions of learning are not excluded in this rapid drive. However, extracting useful information from these stored data has proved extremely challenging (Kumar, 2014). There is need for a robust algorithms that can effectively explored data. Although, the feed-forward neural network is capable of modelling the target attribute as a nonlinear combination of the predictor attributes, but, it sometimes saddled with the problem of over-fitting. This usually affect its outputs especially when simulated with sets of untrained data. It is therefore necessary, to introduce an enhancement to this technique in order to strengthen its predictive capabilities.

Learning techniques such as artificial neural networks have some common challenges in exploring data for prediction purposes. For instance, neural network techniques predict the expected output if properly configured and trained for a number of iterations. Similarly, the decision tree algorithm classifies data based on some class labels on which it is trained. Each of these techniques performs only one main task on the data they explored, i.e. they predict the target. For an efficient exploration of students' data, a robust algorithm that is capable of classifying, summarizing and predict through inference, is desirable. Specifically, this research addressed the following problem/questions:

1. Feed-forward Neural Networks has a very high processing capability which makes it suitable for building of predictive model from data. The problem of over-

fitting sometimes affects the predictive model created using this technique. *How can this technique be enhanced to improve its performance and its ability to generalize*?

2. For the purpose of decision making, an algorithm exploring students' data should be able to summarize, classify and make useful inferences; machine learning algorithms are not dynamic enough to simultaneously handle these enormous tasks. They also tend to decrease in accuracy while fitting model from very large data. To solve this problem requires proposing an algorithm that can explore educational data much better.

3. Huge data are usually captured during the admission process in respect of each student in the institutions of learning; admissions are offered to prospective students based on the captured data. These data are usually neglected after the students have been offered admission and subsequently enrolled in their various programmes. *How can this data be explored further to unveil the usefulness embedded in them*?

1.3 Research Aim and Objectives

This research was aimed at enhancing the performance of feed-forward neural network and to develop an efficient rule-based algorithm, with a view to implementing both to unveil the patterns embedded in the students' data. The historical data explored with the proposed approaches were intended to unveil the background information relating to the students' academic performance. In order to achieve this central goal, some key objectives have been formulated as follows:

1. To analyse the existing studies on predictive modelling of students' academic performance in order to identify the strengths/weaknesses of the approaches earlier proposed.

2. To propose an enhancement for the data division procedures in feed-forward neural network structure for better performance; and to compute the Mean Absolute Error in order to measure the accuracy of the network predictive models created.

3. To develop and implement a Rule-Based Algorithm that is capable of exploring students' data effectively; and to measure the accuracy of this approach in predicting the students' academic performance.

4. To compare the efficiency of the proposed RBA to some existing techniques reported in the literature for the predictive modeling of students' academic performance.

1.4 The Significance of the Research and Scope

This research has a number of significance that can be attributed to it which make the research paramount. The following are some of the significance of this research:

1. The enhancement of the feed-forward neural network proposed in this research prevents over-fitting of the predictive model created using this technique. This improves the performance of the network model.

2. The rule-based algorithm proposed in this research efficiently explores students' data and gives accurate results. Being able to identify the likely victim of drop-out student or those that are prone to over stay on their programme of studies would help to target intervention programmes directly to those students seriously in need. The gesture can also offer an exponential increase of graduated students with good grades.

3. Assessing the prior academic achievement of students and predicting their future performance would offer useful information and numerous opportunities for instructors and decision makers on how to improve their quality of services and for optimal distribution of available teaching resources to curb students' failure.

4. Having knowledge about the students' performance would guide the lecturers on the appropriate standard method of instruction to adopt. If the lecturer categorize all the students to be of equal level of knowledge, by adhering to a very high standard method of giving instructions, students may not be able to understand what they are being taught. On the other hand, if the standard of teaching is kept below the acceptable standard, students may not take the subject seriously and consequently; it may degrade the lecturer's efficiency.

The predictive modelling of students' academic performance focuses on the academic and demographic achievement of the newly enrolled undergraduate students only. The approaches proposed for the model construction are the enhanced feed-forward neural networks and the rule-based algorithm. The creation of models is limited to the past students' data of the newly enrolled students.

1.5 Contributions to Knowledge

This research has contributed to the build on knowledge in Data Mining. The specific contributions of this research are:

1. The enhancement proposed for the feed-forward neural networks in this research, where the partitioning of the data for training was made to be dynamic, alleviate the problem of over-fitting. The enhancement has resulted in a more accurate network predictive model that generalizes much better.

2. This research proposed a Rule-Based Algorithm, which was found to be effective in creating a predictive model of students' performance. The proposed algorithm was tested and comparison was made with some learning algorithms that are known for creating model that perform similar tasks. Findings from this research show that, the proposed RBA is efficient and can be used as an alternative algorithm in the exploration of students' data for prediction purposes.

3. Data pre-processing is an important stage in the data mining process, this research uses an efficient method in transforming the historical data of the newly enrolled undergraduate students. The task reveals the patterns embedded in the data and structure them in a suitable format for mining.

4. This research also shows how the unveiled patterns from the students' data can be efficiently modelled, to predict the future academic performance of the newly enrolled undergraduate students using the techniques of data mining. The revealed information can be of benefit to institution's administrators for decision making.

1.6 Thesis Organization

This thesis is divided into six chapters. The rest chapters are organized as follows:

Chapter 2: The stages involved in the exploration of data using the techniques of data mining and several state-of-the-art techniques including their drawbacks are discussed. The related literature on the building of predictive models from students' data, some of the free software tools that implements several learning techniques are discussed. Also, the discussion in this chapter encompasses a number of some proprietary tools that are well known and gives satisfactory model that handles data mining tasks. A number of techniques proposed in the literature for improving the performance of neural network techniques are also reviewed in this chapter.

Chapter 3: Two approaches are proposed in order to achieve the aim of this research. For clarity sake, the techniques proposed is plitted between chapters 3 and 4. This chapter presents the enhancement proposed for data partitioning function in feed-forward neural network. This chapter discusses the experiments carried out for building of network predictive models from students' data, using both the existing FNN and the proposed enhanced FNN; the experiments carried out in this chapter implements two separate data sets of different sizes. This chapter also shows the computation for measuring the accuracy of a numeric prediction using Mean Absolute Error.

Chapter 4: The second technique used in this research is illustrated in this chapter. The Rule-Based Algorithm proposed and implemented for a more efficient exploration of students' data is represented and discussed in this chapter. The efficiency of the technique is determined by measuring the accuracy of the generated output. Explanation is given in the section of the algorithm, and a sequence diagram that illustrates users interaction with the proposed approach is provided.

Chapter 5: The training performance of all the eight network models created are represented in this chapter. Also, the results of evaluating each model with a view to revealing the effectiveness of each model are represented and discussed in this chapter. In order to further bench-mark the proposed techniques in this research against others, comparisons were made with some state-of-the-art techniques that are well reported for the exploration of data for prediction purposes. The comparisons were based on three metrics.

Chapter 6: This thesis is concluded in this chapter. The significant contributions of this research to knowledge are further discussed. Also discussed in this chapter is the limitations of this research; the applicability of the proposed models; and the direction for further research were also recommended at the end of this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the steps that should be followed in the process of data mining are reviewed. The review in this chapter also encompasses some of the algorithms that are well reported in the literature for the exploration of students' data for prediction purposes. A number of data mining techniques are discussed, specifically, some techniques reviewed in model creation include: Neural Networks, Decision Tree, Naïve Bayesian and Fuzzy Logic. Some free software tools and proprietary tools that implement several data mining techniques are also discussed. This chapter also analyzed some of the related research on modelling of students' academic performance for prediction as reported in the literature.

2.2 Algorithm

An algorithm is a procedure to accomplish a specific task (Skiena, 2008). An algorithm takes any of the possible input instances and transforms it to the desired output. Usually, an algorithmic problem is specified by describing the instances it must work on. The problem solved in this research used a number of procedures to accomplish some specific tasks. Sorting is particularly important when the output of the processed data is to be presented. The concept of sorting algorithm which is also implemented in this research, allows the sorting of scores achieved by the students. For instance, the sorting algorithm takes a sequence of *n* keys say, a_2, a_1, a_3…, a_n. Reordering of this sequence in ascending order would make it to become a sorted sequence of the form: $a'_1 \leq a'_2 \leq \ldots \leq a'_{n-1} \leq a'_n$. Numerically, 1, 7, 3, 5, 10, 6, 8, 9 becomes 1,3,5,6,7,8,9,10 in ascending order and 10,9,8,7,6,5,3,1 in descending order.

Conventional programs such as JAVA, PHP and others, process data based on a clear algorithm, or in other words, a series of well-defined step-by-step operations. An algorithm usually performs operation sequentially, and it always provide an exact solution (Negnevitsky, 2011).

It is always important to ensure that an algorithm is free of ambiguities. A good algorithm should follow a sequential order that can lead to the solution to the problem at hand. A number of vital reasons given for analysing an algorithm as stated in (Sedgewick & Flajolet, 2013), is to discover its characteristics. Such evaluation would reveal its suitability and it can be compared with other algorithms for the same application. The rule-based algorithm proposed in this research is compared with the decision tree algorithm, neural networks learning algorithm and fuzzy logic techniques, since they all model data for prediction purposes.

2.3 Data Mining Technique and Stages in Creating Prediction Models

A model can be described as a simple representation of a more complicated system and could be static or dynamic (Wu & Coggeshall, 2012). Using the technique of data mining, models are constructed or built from data, in order to make prediction or description. A model may be created for the purpose of classifying data so that, similar data would be grouped and dissimilar ones would be clearly separated as in clustering (Lopez et al., 2012). Also, a model can be created to predict the sets of numbers based on the trained numerical data (regression); as in neural network (Naik & Ragothaman, 2004) .

The present research, thus, focuses on building of predictive model from the historical data of undergraduate students. The basis for building a predictive model is basically aimed at predicting for the new record, object, or other dependent variables assumed by the target attribute. Essentially, it is the task of building a model for the target variable as a function of the explanatory variables (Kumar, 2014). Modelling of data for prediction purposes as reported in (Kantardzic, 2011), involves using some variables or fields in the data set to predict unknown or other variables of interest. The techniques of data mining follow a number of steps, there is need to follow these steps sequentially in order to build a desire model from the data. But in an effort to evaluate such model, only such similar data can be used to determine the efficiency of the model created.

The detailed steps involved in the creation of predictive model are enumerated in the subsequent subsections. As shown in the steps, data mining itself is an important stage in the mining process. The analyses of the processes that needs to be followed in the mining of data are discussed as follows:

2.3.1 The Dataset

The data source from which useful knowledge is to be unveiled might be stored in relational databases such as Oracle, MySQL, Access, etc. and at times, it might be from a single source. Data from different sources may also need to be merged into a single repository called a data warehouse. Data warehouse retrieves and periodically consolidates data that is stored in different source systems into a dimensional or normalized data store (Rainardi, 2008).

Integrating files that is sourced from heterogeneous databases sometimes becomes inevitable in order to have the full data in a single location for further processing. The original data may contain several thousands of records; these set of records should then be carefully examined to identify variables that has predictive relevance. Usually, with the knowledge of domain experts, the right selection of most useful attributes can be achieved.

Some of the data used for the prediction of students' academic performance as reported in the literature include: continuous assessment data, enrolment data (Lye et al., 2010; Sharabiani et al., 2014), data from previous semester examinations (Chen & Do, 2014), demographic information such as Age, and other students' personal characteristics (Tan & Shao, 2015). Although, the newly enrolled students do not have continuous or previous semester results, therefore, their academic performance can only be based on their academic antecedents. Most importantly, the data explored for prediction of students' academic performance must have direct bearing with the students. It should also be supported by the domain expert knowledge and some of the predictive variables are expected to relate to what has been reported in the literature.

The enormous proliferation of very large databases in today's companies and scientific institutions makes it necessary for machine learning algorithms to operate on massive datasets. Two separate dimensions become critical when any algorithm is applied to very large datasets: space and time. In an instance where the data is so large that it cannot be held in main memory, this does not pose problem as far as the learning scheme works in an incremental fashion.

A useful approach in a rare case like this, is to process one instance at a time when generating the model; an instance can be read from the input file, the model can be updated, the next instance can be read, and so on without ever holding more than one training instance in main memory. This type of learning is referred to as data stream learning (Witten et al., 2011).

2.3.2 Attribute Selection

Data set that is of high dimension contain many attributes that are redundant or irrelevant for prediction. This can serve as impediments to unveiling the useful knowledge and patterns in the data using the technique of data mining (Jin et al., 2012). For instance, study in (DeBerard et al., 2004) identified some features that if present in student's life may responsible for non-completion of studies or be a victim of students in the retrenched categories.

Health and psychosocial variables such as smoking, drinking, health-related quality of life, social support, and maladaptive coping strategies are factors related to retention. Other predictive attributes reported in the literature includes: gender, perceived stress; time management; religious habits; number of hours worked per week (Trockel et al., 2000). However, the methods used to collect these variables are subjective; therefore, they are not reliable as a predictive attributes. In the study reported in (McKenzie & Schweitzer, 2001), the previous academic performance was identified as the most significant predictor of students' performance in the university. Attributes in this category and students' age were used in several previous studies, see (Oladokun et al., 2008; Tan & Shao, 2015).

Splitting algorithms such as decision tree learners are cleverly designed to choose the best attribute that should be used for splitting at each node. The reason is subtle as one proceeds further down the tree. The data that is available to help in making the selection decision become less and less. At some point, with little data, the random attribute will look good just by chance, this is because the number of nodes at each level increases exponentially with depth, the chance of the rogue attribute looking good somewhere along the frontier multiplies up as the tree deepens (Witten et al., 2011) . The real problem being pointed here is that one will inevitably reach depths at which only a small amount of data is available for attribute selection.

Although, study in (Thiele et al., 2014) is not in support of depending on school qualification alone as predictors. The study opined that, these alone have limited prospective, as predictors of academic potential and other promising socio-demographic attributes should also be considered. Also, study reported in (Fakeye, 2014) shows that, the level of proficiency in English language by high school student, has a significant positive relationship with their overall academic achievement. English language is one of the high school results considered as predictors in this research.

Due to the negative effect of irrelevant attributes on most machine learning schemes, it is common to precede learning with an attribute selection stage that strives to eliminate all, but the most relevant attributes. The best way to select relevant attributes is manually, based on a deep understanding of the learning problem and what the attributes actually mean (Witten et al., 2011).

Also, in a related statement made on attribute selection, study in (Forman, 2003), argued that, no degree of clever induction can make up for a lack of predictive signal in the input features. However, it is also possible to find automatic methods to be useful. Three well reported methods for attributes selection in the literature includes: filter (Nie, 2005); wrapper (Maldonado & Weber, 2009) and hybrid technique (Jin et al., 2012).

2.3.3 Data Pre-Processing

Data set can be viewed as a collection of data objects; other names by which data objects is being addressed are: record, point, vector, pattern, event, case, sample, observation or entity (Kumar, 2014). Generally, the first task that should be performed on a data set is to do some simple data quality examination. This first exploration exercise would reveal the nature of the variation and quality of the data set. This is because, for many real-world application of data mining, most especially when there are huge amounts of data, the subset of cases with complete data may be relatively small.

Most often, a particular algorithm used for building an intelligent system requires a particular type of data. For instance, using the techniques of neural networks, only a numeric data is acceptable for model creation. For better training, it may be necessary to have all the data points normalized, such that, they fall within certain range of values, say 0 and 1. To avoid assigning too much weight to those features that appears to have larger values.

Pre-processing of data prior to exploration is an important task; generally, data are always far from being perfect. Real-world data are susceptible to noisy, missing and inconsistency. Heterogeneous source as identified in (Rotshtein & Rakytyanska, 2012), is one of the factors responsible for unclean data. Notwithstanding, no matter the algorithm to be used to fit models from data, study proposed in (Linoff & Berry, 2011) opined that, the issue of incompatible data, inconsistent data and missing data must be resolved first. This is to avoid likely problems that may associate with the data, which may negatively affect the resulting output of the data analysis.

2.3.4 Transformation/Normalization

Irrespective of the type of model to be created, this step is inevitable. One particular type of transformation is normalization of variables, and other widely used transformation is discretization, which is the division of continuous variables into classes (Tuffery, 2011). Another fairly common transformation technique is one in which the original variables are replaced with their factors, otherwise referred to as factor analysis.

Normalization of data takes different forms, there may be need to scale the values to a specific range, for instance, [-1,1], or [0,1]. This is particularly necessary most especially in order to improve the accuracy and efficiency of mining algorithms involving distance measurements (Rotshtein & Rakytyanska, 2012).

If the values are not normalized, there is every tendency for the distance measures of smaller values to overweight those features that have larger values. There are many ways of normalizing data, some of the effective normalization techniques are listed in (Kantardzic, 2011) as decimal scaling, min-max and standard deviation.

2.3.5 Data Mining

This is the step at which the algorithm for data explorations is implemented. Data mining is an applied area of science (Pechenizkiy et al., 2008) and a key step in the mining process. The tremendous amount of data being regularly captured and stored in large repositories has far exceeded the ability of human to successfully process without a robust algorithm. In order to simplify this task, efforts have been made to develop expert system and knowledge-based technologies, which rely mainly on users or domain experts. This approach manually input knowledge into knowledge bases (Rotshtein & Rakytyanska, 2012). Unfortunately, the manual knowledge input procedure is prone to errors, consumes much time, costly and biased. The widening gaps that exist between data and information, necessitates the development of data mining techniques that can effectively turn data to knowledge.

The data mining tasks are generally divided into two major categories (Kumar, 2014); one of them is predictive tasks. A model with the objective of these tasks is to predict the value of target attribute based on the values of independent attributes. In the present research for instance, the academic performance of students are predicted based on the predictive attributes identified in their previous historic data. The attribute to be predicted is usually referred to as the target or dependent variable, while the attributes used for making the predictions are known as the independent variables.

Data mining has its origin in various disciplines, and as reported in (Kantardzic, 2011), the two most important are statistics and machine learning. Figure 2.1 summarizes those disciplines through which data mining originates.

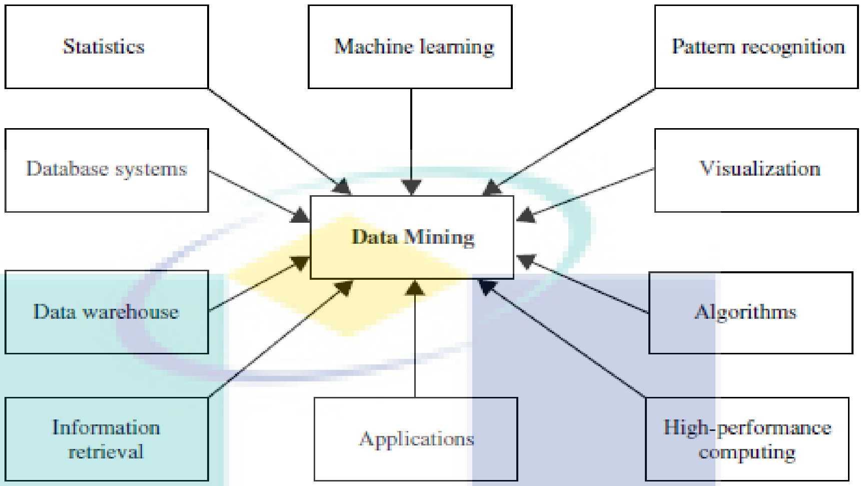

Figure 2.1. Data mining adopts techniques from many domains.

Source: (Rotshtein & Rakytyanska, 2012)

As revealed in the figure, data mining concept brings together techniques from machine learning, pattern recognition, statistics, databases and visualization, to address the extraction of information from large databases. The predictive technique used in data mining can be divided into two major operations: classification and regression (Tuffery, 2011). What distinguishes the two is the nature of the dependent variable, this is qualitative in the case of classification, but continuous in the case of regression. Both operations are explored in this research. The main objective of both operations is to estimate the value of a variable otherwise referred to as dependent, target or response. The independent variables are sometimes referred to as explanatory or exogenous variables.

Classification is the task of assigning objects to one of several predefined categories (Kumar, 2014). Prediction that explore patterns in data for the purpose of classification capture relations of 3 input variables, comprising of 6 numeric values each, say $x_1, \ldots, x_r$ with target variables which must be of equal numeric values, say

$a_1, \ldots, a_f$ here, f = 6. This is illustrated in Table 2.1 and as shown in this table, the three input data are arranged against the expected outputs (target). Those data to be trained are arranged first, while the target data occupies the last column.

Table 2.1
Arrangement of Data for Training

Input1	Input2	Input3	Target
x_1	x_7	x_{13}	a_1
x_2	x_8	x_{14}	a_2
x_3	x_9	x_{15}	a_3
x_4	x_{10}	x_{16}	a_4
x_5	x_{11}	x_{17}	a_5
x_6	x_{12}	x_{18}	a_6

During the training process, the training algorithm tries to establish relationship between the input data and the target. This is referred to as supervised learning technique as the inputs data are being trained to mimic the target data. At the end of the training, the predicted outputs generated is expected to be very close in value to the actual value (target), the difference is the error.

2.3.6 Evaluation

Evaluation is the key to making real progress in data mining as there is always the need to examine the resulting outputs of a model for correctness (Witten et al., 2011). A number of evaluation measures are available to determine the present of error in a model, this depends on the type of model i.e what the model does.

In order to evaluate the performance of a model whose target is a continuous value, several alternative measures presented in Table 2.2 can be used to evaluate the success of the numeric predictions. The predicted values on the test instances are p_1, p_2, …, p_n; taking as an example, the value of p as 0.4, 0.6, 0.8, 0.9…, while the actual values are a_1, a_2, …, a_n. Examples of the actual value may be 0.4, 0.58, 0.7, 0.89. It should be noted that p_i illustrates a different meaning here from what it meant in the classifier model, where it was referred to as the probability that a particular prediction was in the *i*th class. In regression model, it refers to the numerical value of the prediction for the *i*th test instance (Witten et al., 2011).

Four of the evaluation measures that can be used to compute the accuracy of numeric prediction as enumerated in (Witten et al., 2011) includes:

1. ***Mean Squared Error*** *(MSE)*: This is the principal and most commonly used measurement; it is sometimes referred to as objective function. The square root is taken to give it the same dimensions as the predicted value itself. Many mathematical techniques such as linear regression use the mean-squared error due to the fact that, it tends to be the easiest measure to manipulate, the mathematicians usually say, "well behaved." The MSE can be used in several instances, but here, it is being used as a performance measure. Generally, most of the performances are easy to calculate, so mean-squared error has no exceptional advantage for this purpose.

2. ***Mean Absolute Error*** *(MAE)*: This is the average of the magnitude of the individual errors regardless of their sign. Mean-squared error tends to exaggerate the effect of outliers in dataset when the prediction error is larger than the others, but the MAE does not have this effect. All sizes of error are treated evenly according to their magnitude. In terms of importance, sometimes it is the *relative* rather than *absolute* error values that may be seen as vital. For example, if a 10% error is equally important whether it is an error of 50 in a prediction of 500 cases or an error of 0.2 in a prediction of 2 cases, then averages of absolute error will be meaningless, the relative errors appears to be appropriate in an instance like this. This effect would be taken into account by using the relative errors in the mean-squared error calculation or the mean absolute error calculation.

3. ***Relative Squared Error*** *(RSE)*: This differs a bit from the previous error measurements. Here, the error is made relative to what it would have been if a simple predictor had been used. The simple predictor in question is just the average of the actual values from the training data, which is denoted by '*a*' in Table 2.1. Thus, relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the default predictor. The root relative squared error is obtained in the obvious way.

4. ***Relative Absolute Error*** *(RAE):* This is simply the total absolute error, with the same kind of normalization. In the relative error measures, the errors are normalized by the error of the simple predictor that predicts average values. These measurements of numeric predictions are further summarized in Table 2.2

Table 2.2
Performance Measure of Numeric Prediction

Evaluating measures	Formula
Mean Square Error	$\frac{(\rho_1 - a_1)^2 + ... + (\rho_n - a_n)^2}{n}$
Mean Absolute Error	$\frac{\lvert\rho_1 - a_1\rvert + ... + \lvert\rho_n - a_n\rvert}{n}$
Relative-Square Error*	$\frac{(\rho_1 - a_1)^2 + ... + (\rho_n - a_n)^2}{(a_1 - \bar{a})^2 + ... + (a_n - \bar{a})^2}$
Relative Absolute Error*	$\frac{\lvert\rho_1 - a_1\rvert + ... + \lvert\rho_n - a_n\rvert}{\lvert a_1 - \bar{a}\rvert + ... + \lvert a_n - \bar{a}\rvert}$

* $\bar{a}$ is the mean value over the training data.

Sometimes, the target attribute may consist of non-numeric data. In such instance, the above listed measures would not be appropriate for measuring the prediction accuracy. There are many evaluation measures suitable for measuring the accuracy in a classification problem. For instance, when a predictive model is created for the purpose of classifying the dataset based on certain class labels, then, any suitable measure among the following evaluation measurement represented in Table 2.3 can be used to determine its accuracy. The target of such data may be categorical or simply comprised of data that are alphabetic, alpha-numeric or sometimes special characters.

Table 2.3
Measure of Class Labels Prediction Model

Measure	Formula
Accuracy, Recognition rate	$\frac{TP+TN}{P+N}$
Error rate, Misclassification rate	$\frac{FP+FN}{P+N}$
Sensitivity, True positive rate, Recall	$\frac{TP}{P}$
Specificity, True negative rate	$\frac{TN}{N}$
Precision	$\frac{TP}{TP+FP}$

Source : (Witten et al., 2011)

In relation to region,

TP represents true positive; TN represents true negative

FP represents false positive; P represents positive; N represents negative;

2.3.7 Knowledge Representation

The last stage of data mining process is knowledge representation. The model outputs can be represented, for instance by using graphs, charts, tables, etc. Visualization and knowledge representation techniques are used to present the mined knowledge and subsequently interpreted, for better understanding. The simplest and most rudimentary way of representing the output of the mined data is to make it just the same as the input, i.e in a tabular format (Witten et al., 2011). What is considered as most important, is to represent the knowledge that is revealed in the most simplest way for someone else to understand.

The steps discussed so far, on the process or stages involved in data mining is further summarized in Figure 2.2.

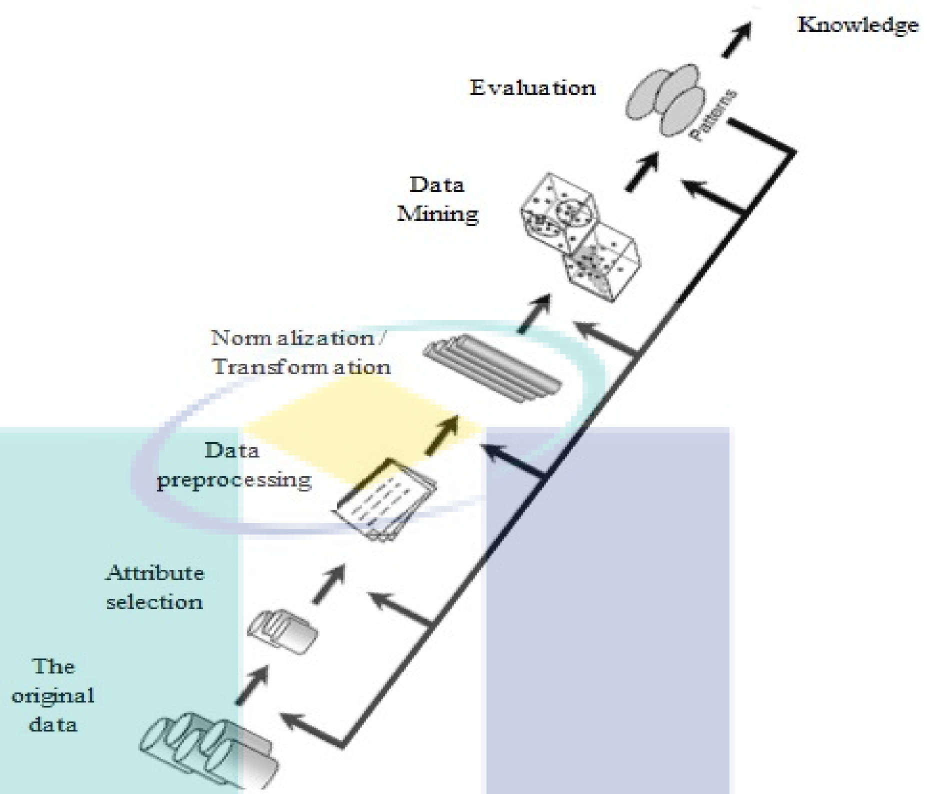

Figure 2.2. The data mining process.

Source: (Baradwaj & Pal, 2011)

2.4 Software Tools for Creating Prediction Models

The last subsection discusses the steps that must be followed while using the data mining techniques to build models. In the present section, the focus is specific on the techniques that can be used to create prediction models. Whether models for students' academic performance prediction or prediction models to achieve other tasks, basically, the same process needs to be followed. What clearly differentiate them is the input data and the output that is expected, target. The interpretations of the predicted outputs in each case also differ.

Although, several techniques are reported in the literature for the creation of prediction models, however, the focus of this thesis is on those techniques that are well known for the creation of prediction model for students' academic performance. Among the well reported techniques in this category include: Neural networks, Decision tree, Fuzzy logic and Naïve Bayesian. These are mainly machine learning techniques except fuzzy logic.

A number of tools have also been developed for the implementation of these techniques, four open source tools that implements several useful algorithms for use in data mining are discussed as follows:

2.4.1 Weka: This is an open source collection of machine learning algorithms suite written in Java for data mining tasks; it also includes neural network capabilities. The acronyms stand for Waikato Environment for Knowledge Analysis. The tool was developed at the University of Waikato in New Zealand. It composed of mainly machine learning algorithm for data mining tasks. Weka provides implementations of learning algorithms that can easily be applied directly to a dataset or a java code can be written to call the algorithm that does the implementation.

The software also includes a variety of tools for transforming datasets, such as the algorithms for discretization. It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating systems and even on a personal digital assistant (Witten et al., 2011). Weka contains tools for data pre-processing, classification, regression, clustering, association rules and visualization.

2.4.2 Rapidminer: RapidMiner and RapidAnalytics provide an integrated environment for all steps of the data mining process. The tool was initiated in 2001, and its first version was called Yet Another Learning Environment (YALE). It provides environment that can be used to extract meaning from dataset. Rapid-I, founded in 2006 is the company behind the open source software solution, RapidMiner and its server version, RapidAnalytics.

RapidMiner is able to read from databases with ease. With this software tool, there are hundreds of machine learning operators to choose from, several helpful pre and post processing operators, descriptive graphic visualizations and many other useful features . Most databases such as MySQL, PostgresSQL, SQL Server, Sybase, Oracle, and Access are supported by this tool (Chisholm, 2013) .

RapidMiner also simplifies the design of data mining processes by a simple drag and drop of boxes. The functional modules also called operators can be moved into the process to define data flows, by simply connecting these boxes. It can also be used to define even complex and nested control flows, and all without the need for coding. The tool also provides integrated environment for text mining and predictive analytics. Rapidminer is used for industrial applications, research, education, training, rapid prototyping and application development.

2.4.3 Knime: The Konstanz information miner is a data analytics platform with a large set of building blocks and third-party tools. KNIME can be used right from the loading of data to a final report. The tool is suitable for the prediction of new values using a previously found model. The software tool is available in four flavours, however, only the Desktop version is open source. The software tool is a user-friendly and comprehensive data analytics framework which offers capabilities for the entire analytical process: Data access, data transformation, powerful predictive analytics, visualization and reporting.

The main views of KNIME give multiple options to explore data, which enable easy visual assembly and interactive execution of data pipeline. From a variety of nodes in KNIME, one can select data sources, data pre-processing steps, modelling building algorithm as well as visualization tools (Berthold et al., 2008).

2.4.4 ML-Flex: This software tool was motivated by a need to classify high-dimensional, heterogeneous data. ML-Flex was written in Java but is capable of interfacing with third-party packages written in other programming languages. It is capable of handling multiple input-data formats and supports a variety of customizations. Execution of ML-Flex revolves around the concept of experimentations, in an experiment, the user specifies one or more sets of independent (predictor) variables and a dependent variable (class) as well as any algorithm(s) that should be applied to the data (Piccolo & Frey, 2012).

All the four software tools discussed are open source. There are some other proprietary software tools found very useful for creating prediction models, these include Oracle Data Mining, STATISTICA Data Miner, SAS Enterprise Miner etc. Other state-of-the-art techniques for the creation of prediction models are discussed in the following sections.

2.5 State of the Art Techniques for Creating Prediction Model

A number of techniques have been reported in the literature for the creation of prediction models, four of these techniques are reviewed here. Models created for Classification or Regression is a data mining models and the two techniques are form of prediction, it is only their target that differs. While classification model has labels as target, regression has continuous values.

2.5.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are viable computational techniques for a wide variety of problems. The networks are capable of learning using training examples, to perform tasks which previously required costly programming (Black & Ertel, 2011). The network has also been described as a model of reasoning based on the human brain (Negnevitsky, 2011). Artificial neuron is the basic building block or processing unit of an artificial neural network (Hassoun, 2008).

The techniques of neural network are widely reported for the creation of a predictive model, where for instance, a number of predictors are properly trained in order to mimic some known outputs. A model has been described as an abstraction of some aspect of a problem (Blaha, 2010). The neural networks can be categorized into several broad techniques that comprised of many network architectures, including multilayer perceptron and normalized radial basis function networks. The network also includes supervised and unsupervised learning. A learning is said to be supervised if the target outputs are provided, while it is unsupervised if the learning technique devoid of the target outputs.

To create a prediction model using ANNs, the first thing to do after collecting the data through which the model is to be fitted is to identify variables that have predictive relevance. There is need to know what is to be predicted which would serve as the target data. Then all the input data that is capable of achieving the target data should also be selected. For neural network to be used effectively for creating a prediction model, the data must be normalized.

Also, the input data set from which a model is to be created is usually arranged in a similar way when using a learning algorithm. Neural network is also not an exception in this pattern of arrangement. All the input data are arranged first, while the target data usually occupies the last field. This is a supervised learning approach, whereby, the input attributes are associated with a specific desired target.

Neural networks have a number of algorithms that can be used for training. The typical algorithm used for training of multilayer perceptron network is back propagation algorithm (Wu & Coggeshall, 2012), represented in Figure 2.3. One of the reasons for training with back propagation algorithm is due to its mathematical tractability and most times, converges to a better solution (Rotshtein & Rakytyanska, 2012).

Setting the data to be trained in this way causes the weight of the inputs to be synthesized gradually. At the end of each iteration, the process of learning is updated. This is to reduce the error between the network predicted output and the corresponding expected output.

For quite a long time, neural networks have proven to be extremely powerful techniques for the data exploration through which models can be created in order to discover previously unknown dependencies and relationships in datasets (Andrews et al., 1995). The error in the course of training is calculated as the difference between the target output and the network output and in order to minimize the average of the sum of these errors, the training algorithm adjusts the weights and biases of the linear network.

In terms of practical application, backpropagation model is the most widely used model. Although, no statistical survey have been conducted to establish this fact, but as reported in (Hassoun, 2008), it is believed that, not less than 90% of commercial and industrial applications of neural networks uses backpropagation or its derivatives. The algorithm is also considered in (Maimon & Rokach, 2008), as the most popular learning paradigm in neural networks applications and it is essentially a gradient steepest descent method.

According to Maimon and Rokach (2008), the idea of steepest descent method is to find the best direction in the multi-dimension error space to move or change the weights so that the objective function is reduced most. The objective function in this context is the Mean Square Error. The partial derivative of the objective function is required with respect to each weight to be computed; this is because, the partial derivative represents the rate of change of the objective function.

There are many training functions that can be used for backpropagation model, one of the training function that has proven to be very fast, and which is used for training in this research is TrainLM. It is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix (Coskun & Yildirim, 2003). The steps involved in backpropagation algorithm are represented in Figure 2.3.

Input: D, a dataset consisting of the training tuples and their associated target values;

L, the learning rate;

Network, a multilayer feed forward network.

Output: A trained network

Method:

1. Initialize all network weights and biases ;
2. **while** terminating condition is not satisfied {
3. **for** each training tuple X in D {
4. **for** each input layer unit j {
5. $O_j = I_j$ // output of an input unit is its actual input value
6. **for** each hidden or output layer unit j {
7. $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
8. $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
9. **for** each unit j in the output layer
10. $Errj = O_j (1-O_j)(T_j-O_j)$; // compute the error
11. **for** each unit j in the hidden layer,
12. $Errj = Oj\,(1-Oj) \sum_k Err_k wj_k$; // compute the error with respect to the next higher layer, k
13. **for** each weight w_{ij} in network {
14. $\Delta w_{ij} = (l)\, Err_j O_i$; // weight increment
15. $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
16. **for** each bias θ_j in network {
17. $\Delta \theta_j = (l)\, Err_j$; // bias increment
18. $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
19. }}

Figure 2.3. Back propagation algorithm.

Source: (Han et al., 2012)

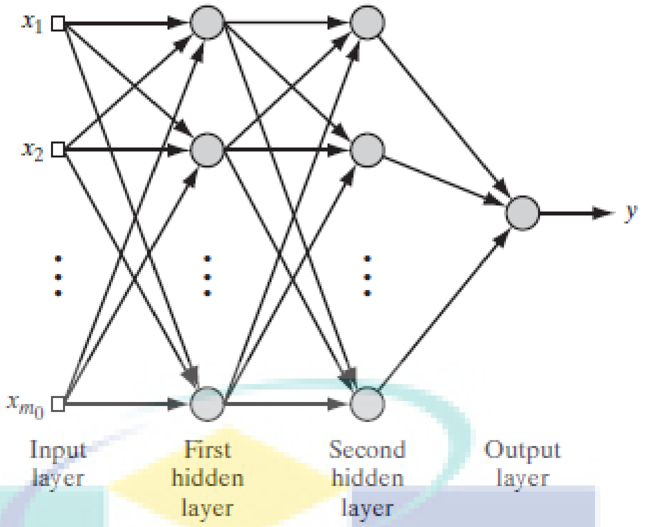

Figure 2.4. Multilayer perceptron neural network architecture.

Source: (Haykin, 2009)

Objective functions perform important roles in the course of training data and as reported in (Wu & Coggeshall, 2012), two commonly used objective functions are the Mean Square Error and Maximum Likelihood (ML). The structure shown in Figure 2.4 represents multilayer perceptron architecture with 2 hidden layers, $x_1 \ldots x_{mo}$ are the inputs, while y is the output.

Although, the basic operations in neural network take place within the hidden layer, but the network still complies with certain general rules for the adaptive training. The gradient descent rule with a minimum term reported in (Wu & Coggeshall, 2012) is represented in the Eq. (2.2). The change in weight during the training process:

$$\Delta w_{ij}(t) = -\eta . \frac{\partial E}{\partial w_{ij}} + \alpha . \Delta w_{ij}(t-1) \qquad (2.2)$$

where E is the objective function, w_{ij} is the weight connecting the node j to i, η is the learning rate, and α is the momentum rate normally chosen between 0 and 1. The $\Delta w_{ij}(t-1)$ is a special case of multistage gradient methods for accelerating convergence. In the process of training a network, some of the activation functions that can be used are: step, sign, sigmoid and linear function. The *step* and *sign* activation functions, is sometimes referred to as hard limit functions. They are mostly used in decision-making neurons for classification and pattern recognition tasks (Negnevitsky, 2011). Back propagation algorithm is usually the algorithm of choice for training of data in neural networks; due to its mathematical tractability and most times, converges to a better solution (Rotshtein & Rakytyanska, 2012).

The techniques of neural networks is widely reported for building of predictive network model from students' data (Chen & Do, 2014; Ibrahim & Rusli, 2007a; Oladokun et al., 2008; Tan & Shao, 2015). The learning algorithm is capable of modelling the class prediction as a nonlinear combination of the inputs. It has proven its suitability in solving several complex tasks, most especially when trained with back propagation algorithm. The neural network method is also non-parametric, i.e the independent variables are not assumed to follow any particular probability distributions. Neural networks can model a wide range of problems, including clustering, classification or prediction (Tuffery, 2011).

Although, the techniques have a proven track record of success for certain specific problem domains, neural network is not without drawbacks. Its black box approach has several consequences, the processing in the hidden layer is opaque (Negnevitsky, 2011), it is therefore difficult, to give explanation on how the inputs transform to the outputs due to its non-explicit nature. This makes troubleshooting so difficult when the network do not work as expected. Other common challenges of this learning technique as reported in (Tuffery, 2011) includes: The impossibility of handling an excessively large number of variables, the considerable risk of over-fitting, if the number of cases is too small with respect to the number of units, the difficulty of using the networks correctly, because the parameters are numerous and hard to control.

2.5.2 Decision Tree

A decision tree is a tree data structure that consists of decision nodes and leaves (Ruggieri, 2002). A leaf specifies a class value. A decision node specifies a test over one of the attributes, which is called the attribute selected at the node.

1. ComputeClassFrequency(T);
2. If OneClass or FewCases
   return a leaf;
   create a decision node N;
3. ForEach Attribute A
   ComputeGain (A);
4. N.test = AttributeWithBestGain;
5. If N.test is continuous
   Find threshold;
6. ForEach T' in the splitting of T
7. If T' is empty
   Child of N is a leaf
   else
8. Child N = FromTree (T');
9. ComputeErrors of N;
   return N.

Figure 2.5. C4.5 Algorithm for decision tree.

Source: (Ruggieri, 2002)

C4.5 is a typical algorithm used for splitting of variables in decision tree, its algorithm is as shown in Figure 2.5.

A decision tree technique is one of the most intuitive and popular data mining methods, especially as it provides explicit rules for classification and copes well with heterogeneous data, missing data and non-linear effects (Tuffery, 2011). The decision trees are on boundary between predictive and descriptive methods since they create their classification by segmenting the population to which they are applied (Tuffery, 2011), thus, they belong to the category of supervised divisive hierarchical methods.

It is used in classification to detect criteria for dividing the individuals of a population into n pre-determined classes and in many cases, n = 2 (Tuffery, 2011).

Tree-based models are widely used for data exploration. Their strength is in simplicity of presentation and always easy to understand. There exist a group of well-known decision tree algorithms, such as ID3, C4.5 and C5.0. These algorithms differ in dealing with discrete and continuous outputs (Wu & Coggeshall, 2012). CART represents an attractive powerful non-parametric technique that generalizes parametric regression models; it allows for nonlinearity and variable interactions without having to specify the structure in advance (Ledolter, 2013).

The application of decision tree algorithm involves tracing a path from the root to a leaf node, which holds the class prediction for that tuple. According to (Rotshtein & Rakytyanska, 2012), decision trees can easily be converted to classification rules. To construct a decision tree, with the intent of dividing the individuals of a population into *n* classes, the variable which best separates the individuals of each class must be chosen. According to the precision criterion (García & Mora, 2011), the choice of the variable and the separation condition on this variable depends on the type of tree. A typical decision tree has the structure represented in Figure 2.6

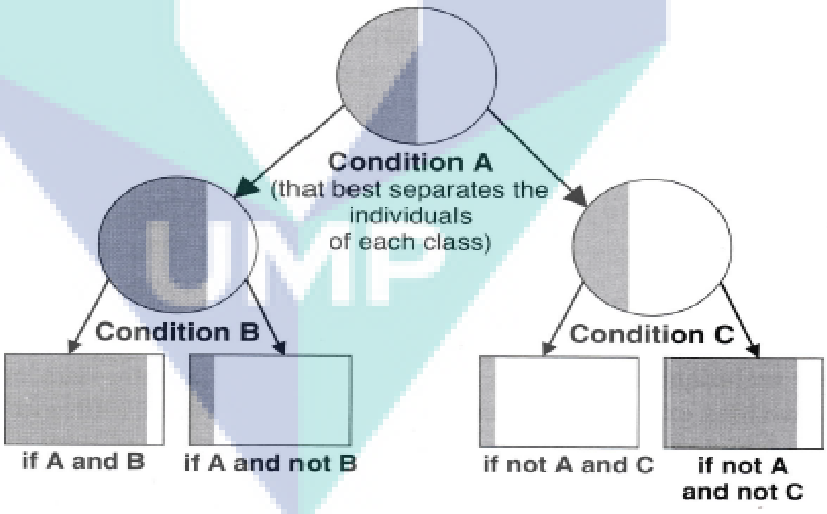

Figure 2.6. A typical decision tree structure.

Source: (Tuffery, 2011)

After building the decision tree, a tree-pruning step can be performed to reduce the size of the decision tree. Decision tree that are too large are susceptible to over-fitting. There is over-fitting when the error on the training set is driven to a very small value, but when new data are presented to the model, the error becomes large (Beale et al., 2010). It means that, the model has learned the trained data, but the learning is not sufficient enough to generalize to new situations.

2.5.3 Naïve Bayesian

A Naïve Bayes as the name suggests, is a classifier that is based on the Bayes theorem. Using this technique derives a conditional relationship among various values. The basics of the theorem can be described as reported in (Kumar, 2014), that:
Given two values X and Y as a pair of random variables, their joint probability, P(X= x, Y= y), refers to the probability that variable X will take on the value x and variable Y will take on the value y. A conditional probability is the probability that a random variable will take on a particular value given that the outcome for another random variable is known (Kumar, 2014). The joint and conditional probabilities for X and Y are related in the following ways:

$$P(X, Y) = P(Y|X)\,P(X) = P(X|Y)\,P(Y) \tag{2.3}$$

The Bayes theorem is derived from equation 2.3 as:

$$P(Y|X) = \frac{P(X \mid Y)\,P(Y)}{P(X)} \tag{2.4}$$

The Bayes theorem is a statistical principle for combining prior knowledge of the classes with new evidence gathered from the data. A naïve Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label y (Kumar, 2014).

The advantage of Bayesian networks is that, they enable knowledge to be represented in graphic form and sometimes, the expression white box is used, as opposed to the black box of the neural network (Tuffery, 2011); they are easy to use and modify, and very suitable for drawing inferences. The technique is reported to be appropriate for the exploration of students’ data in (Baradwaj & Pal, 2012).

The Naïve Bayes classifier has some drawbacks. According to (Kumar, 2014), the correlated attributes can degrade the performance of naïve Bayes classifiers because, the conditional independence assumption no longer holds for such attributes as it may seem too rigid.

2.5.4 Fuzzy Logic

Fuzzy logic is based on the theory of fuzzy set. The theory of fuzzy set was introduced by Professor Lofti Zadeh in 1965, as a means of representing and manipulating data that was not precise, but rather fuzzy (Fullér, 2013). Fuzzy logic attempts to model human's sense of words, decision making and common sense as it is leading to more human intelligent machines. Sometimes, novice compare the fuzzy set theory and the theory of probability, although, both techniques revolves around uncertainty, but in reality, both theories treat the concept differently (Rotshtein & Rakytyanska, 2012).

Fuzzy logic is particularly good for knowledge representation, although, the fuzzy systems generally lack learning ability (Taylan & Karagözoğlu, 2009), it is not robust in relation to the topological changes of the system, this is one of its drawbacks. A fuzzy set can simply be defined as a set with fuzzy boundaries (Negnevitsky, 2011). The development of fuzzy logic was motivated in large measure by the need for a conceptual framework which can address the issue of uncertainty and lexical imprecision. This technique does not use a specific algorithm for problem solving but models data based on the set theorem. It is a very useful modelling soft-computing technique and require fewer rules (Negnevitsky, 2011). The technique deals with vague, imprecise and uncertainty of knowledge.

The reasoning by fuzzy system is usually of two parts, the IF part of the rule (antecedent) and the THEN part of the rule (consequent). Building fuzzy expert systems is an interactive process that involves: defining fuzzy sets and fuzzy rules, evaluating and then turning the system to meet the specified requirements. Fuzzy logic has more than one meaning. Looking at the technique narrowly, it can be seen as a logical system, which extends from multivalued logic. Looking at the concept in a wider sense, fuzzy logic is almost synonymous with the theory of fuzzy sets (MathWorks, 2015a); a theory which relates to the classes of objects that does not have a sharp boundary.

The fuzzy logic membership is a matter of degree. Thus, a proposition is not either true or false, but may be partly true (or partly false) to a particular degree (Negnevitsky, 2011). In fuzzy set, the degree of membership is usually taken as a real number in the interval [0,1]. But the Boolean logic is of two valued logic, where an element either belong to a set with a membership of 1, and in case it does not belongs to the set, then its membership is 0.

The set X is referred to as the universe of discourse Classical reasoning uses the inference rules on implications. This is extended by fuzzy logic to their generalized versions, in which the antecedents and consequents of fuzzy implications are fuzzy sets, usually referred to as approximate reasoning (Shinghal, 2013). In order to represent a fuzzy set in a computer, the membership function must be determined first; this may require application of several knowledge acquisitions. Knowledge may be acquired from single or multiple experts within the domain of the fuzzy set.

Fuzzy logic technique can be used to model students' data for prediction purposes. The technique has a set of mathematical principles for knowledge representation based on degree of membership. The concept deals with reasoning on a higher level, using the linguistic information acquired from domain experts. The study in (Vieira et al., 2004), is of the view that, the fuzzy logic technique has the capacity to represent the inherent uncertainties of the human knowledge with linguistic variables. Due to natural rule representations, the resulting outputs of using this technique are always easy to interpret.

2.6 Rules Generating Techniques for Creating Prediction Model

A number of rule generating techniques have been very successful in solving several classification problems as the approach can be well easy to interpret and understand (Klawonn et al., 1995). A set of rules provided by domain expert knowledge may be stored in a database or embedded within the codes. If stored in a database, then there may be need to create a kind of look up statements in the code to access the table file each time certain decisions is to be taken.

In order to ensure that the steps taken in performing a task is well understood, some existing techniques makes it possible to generate rules that lead to their solution. Prominent among these techniques are: Decision tree, Fuzzy logic and Neuro-fuzzy. For instance, study in (Haimowitz et al., 1999) created a set of rules that matches new records and existing records of different set of people in a large database, in the process, CART was implemented. CART is a version of decision tree algorithm and the acronym stands for Classification and Regression Technique.

Generating rules from the techniques that are listed earlier, involves the use of IF statement, THEN statements. Fuzzy logic is a knowledge representation technique that is based on Set theorem, and as reported in (Klir & Yuan, 1995), rules are generated based on the expert's knowledge and the use of linguistic variables; illustration of rule statements is:

"If height = tall then trouser length = long"
" If fever = high and headache = severe then disease = viral"

In relation to students' academic performance, the IF statement is tested on the students' achievement (scores), the range of values in which the score belong determines where it should be assigned in terms of performance by the THEN part of the rule.

The process of generating rules in Neuro-fuzzy and Fuzzy logic is similar, what differentiates both approaches is the amalgamation of neural network technique that makes learning possible through adaptation.

Also, once the fuzzy sets and rules are set up, they do not change, the fuzzy logic technique does not learn from new inputs (Klir & Yuan, 1995), this is why it is not capable of generalizing when exposed to set of new data.

Having recognized the importance of rules in giving a clear understanding of how input transform to output, the present research explored students' past data based on a number of rules. The past academic antecedents of the students are computed and through these rules, their performance was predicted. The prediction is made strictly for management and prevention of risk, to take immediate actions at the earliest stage of their studentship at ensuring their success.

2.7 Reviews of the Related Research

The techniques discussed so far, are widely used for predictive modelling of students' data. Studies reported in the literature have shown that, neural network techniques standout in the modelling of numeric data for prediction purposes. This research has, therefore, chosen neural networks for the predictive modelling of students' academic performance. This is achieved by unveiling useful knowledge from students' past data. The main reason for the choice of this technique lies in its capability and efficient modelling of the class prediction as a nonlinear combination of the inputs. While decision techniques earlier discussed handles classification very well, neural network techniques can be used for classification (Romero et al., 2008), regression (Baradwaj & Pal, 2011) and clustering (Van der Maaten & Hinton, 2008).

In this research, an enhancement on this technique was first proposed prior to using it to fits predictive model from the students' data. In the next subsection, the enhancement proposed for neural networks as reported in some earlier studies are reviewed.

2.7.1 Improving the Performance of Neural Network Techniques

A number of researchers have made some useful efforts at improving the performance of models created using the neural networks techniques. In this thesis, some of their proposed work is reviewed. Improving the neural network performance has since caught the attention of many researchers, some proposed studies proposed in this area includes: improving the performance of network on new inputs (Cohn et al., 1994), enhancing the network topology (Stanley & Miikkulainen, 2002), improvement on algorithms for network training (Ampazis & Perantonis, 2002), Improving neural networks by preventing co-adaptation of feature detectors (Hinton et al., 2012) etc. These are discussed one after the other in the subsequent paragraphs.

The design of neural network connectivity to give a satisfactory performance is a complex task; this most times involves trying several numbers of neurons in the hidden layer(s). However, the use of pruning algorithm proposed in (Thimm & Fiesler, 1995) and constructive algorithms for structure learning in feed-forward neural network proposed in (Kwork & Yeung, 1999) provides some elaborate and useful methods on how network can be configured to achieve some level of satisfactory performance.

The performance of Feed-Forward Neural Network (FNN) is enhanced in (Erkaymaz et al., 2014) using the small-world topology and a similar network but with zero rewiring was tested using the same dataset. The comparison of the results in both cases shows that the small-world topology improved the performance of FNN. Earlier studies of supervised learning in a multi-layered FNN (Simard et al., 2005), revealed that, architecture based on small-world reduces both learning error and training time when compared to regular network.

The study proposed in (Lin et al., 2009) presents a method for improving the generalization performance of a radial basis function (RBF) neural network. RBF is one of the network structures in neural network. The enhancement proposed for the RBF, a statistical linear regression technique was based on the orthogonal least squares algorithm. The performance of the network model was reported to have improved using a bootstrap technique. Consequently, the result of simulating the improved network model demonstrates a significant improvement. Especially, as regards the generalization performance of the algorithm the study proposed over the existing methods.

The computer –aided optimal design that was proposed in (Issanchou & Gauchi, 2008), aimed at improving the neural network predictive ability. The study designed a strategy based on statistical concepts in order to improve FNN generalization. The study carried out Monte Carlo-simulation based method to examine the usefulness of the design approach in the context of FNN. Study in (Yam & Chow, 2001) proposed an enhancement of the FNN speed, the approach aimed at determining the optimal bias and magnitude of initial weight vectors based on multidimensional geometry. The study was validated through simulations and comparative study.

The generalization of the predictive model was the focus in the study proposed in (Cohn et al., 1994). The work proposed an improvement for the neural network with active learning. The study described formalism for active learning concept which is otherwise known as selective sampling. The study also reported the details required for the implementation of this technique in order to bring about the desired improvements in the performance of the network model.

In an efforts to improve the predictive ability of the learning machine and in particular, the neural network, a modified particle swarm optimization algorithm was proposed in (Fei et al., 2012). The proposed study was to select the input weights and hidden biases of single-hidden-layer feed-forward neural networks with a view to improving the model performance. The study was reported to have better generalization performance. In the study proposed in (Zhao et al., 2009), the authors argued that the random input weight selection using the Extreme Learning Machine (DeBerard et al.) algorithm may lead to an ill-conditioned problem, for which solutions will be numerically unstable. As an alternative, the study proposed selection algorithm for an ELM with linear hidden neurons and the resulting output was reported to have maintained accuracy.

Study in (Narayan et al., 1996) proposed an enhancement to the neural network structure. The study has the motives of reducing the time required to train multi-layered perceptron networks; as time is always a big consideration when one is dealing with big datasets. The technique was designed for implementation within the framework of a back propagation model.

The present research proposed an enhancement that is capable of alleviating the problem of over-fitting. The enhancement achieves a predictive model that generalizes well and maintains accurate performance when faced with set of untrained data. In addition, this research develops a rule-based algorithm for efficient exploration of students' data. The algorithm classifies, summarizes and makes useful inference needed for students' performance prediction.

2.7.2 Predictive Modelling of Students' Academic Performance

The use of learning algorithm for modelling of students' academic performance is well reported in the literature (Arsad et al., 2014; Arsad et al., 2013; Chen & Do, 2014; Ibrahim & Rusli, 2007a; Oladokun et al., 2008; Tan & Shao, 2015; Win & Miller, 2005). In the institutions of learning, it is a good idea to periodically evaluate the retention rate of students. This would enable the institution's administrators to understand why students are dropping out or why the institution is having significant variability in enrolment patterns (Brown, 2009).

Findings from such evaluation would guide the institution on the proper steps to take in order to address whatever anomalies identified. It is very rare, if not impossible, for all students in an institution to be of equal talent. What is obtainable as revealed in the analysis of the reviewed studies, is the variation in students' knowledge as regards their academic performance.

The use of learning algorithm for fitting predictive models from students' data is well reported in the education data mining research. The study proposed in (Lye et al., 2010), used the back propagation algorithm for the training of some students' data. The study focused on developing a model for the prediction of students' performance in their final year examination. The study and several other similar studies used Back propagation algorithm for the training of the data sets.

Also, a comparative study for the use of some selected machine learning algorithm to determine their suitability of fitting models from data is reported in (Ibrahim & Rusli, 2007b). Findings from the study show that, neural network model gives better prediction outputs. The learning techniques of neural networks through which new patterns and functional dependencies are learnt has been described in (Suh, 2012), as model free estimator. This is because, ANNs do not use a mathematical model of how a system's output depends on its input.

In a case study reported in (Oladokun et al., 2008), the study used the neural network technique for construction of models to predict the students' performance in an engineering course. Confusion matrix generated for the predicted outputs show that, an accuracy of about 70% was achieved.

Also, the study proposed in (Arsad et al., 2012) for the prediction of students' academic performance used the neural network techniques. The study focused on the Electrical Engineering degree students; the study based its input data on three attributes and the outcomes of the study reported a direct correlation between the students' results for core subjects at semester one, with the final overall academic performance irrespective of their gender.

The decision tree is another useful technique for model construction. The study carried out using the decision tree technique as reported in (Bunkar et al., 2012) predict the performance of students with some acceptable level of accuracy. Findings from the study show that, it is one of the methods that can be reliably used for academic performance prediction. The technique generate rules which makes the resulting outputs easy to interpret. The rules generated through this approach can be implemented using any suitable programming language. However, there is a tendency to experience a reduction in accuracy as the tree grows with many branches.

Decision tree is reported to have been used to predict the student's performance in web-based e-learning systems in (Chan, 2007). In order to improve the performance of a predictive model, a combination of fuzzy logic and decision tree were proposed for modelling of data in (Damez et al., 2005). Rules were generated; the interesting part of using decision tree is the clarity of statements on which the rules were based. The sequence through each node is easy to follow and most times, self-explanatory.

Also, the study proposed in (Gu & Zhou, 2012) for student performance prediction was based on improved C4. 5 decision tree algorithm. The proposed work analyses some students' data and examination scores; the study creates the classification model and generate rules, which help to predict the prospect of student's scores. The predictive attributes used for construction of the classification model included: student type, admission type, average exam score, entrance exam scores and bonus scores. C4.5 algorithm is an efficient algorithm as it can split the node more than two. Unlike CART, that is restricted to two i.e Yes or No.

The study in (Taylan & Karagözoğlu, 2009) introduced a systematic approach for the design of a fuzzy inference system which was based on a class of neural networks to assess the students' academic performance. The study considers six input attributes which comprises of scores from: quiz, major, midterm, final, performance Appraisal and survey. The study was reported to have been carried out in order to give the necessary feedback on the information regarding the effectiveness of teaching and learning.

Bayesian network is another technique the has been reported for prediction purposes. The technique of Bayesian network classifier was applied in the study proposed in (Sundar, 2013). The study compared a number of classifier algorithms and explored some predictive attributes such as the previous semester marks, internal marks, performance in the seminars, assignment, attendance etc., to construct a predictive model. The Bayesian network classifier is a statistical approach and as proved useful and efficient.

Also, the study proposed in (García & Mora, 2011) was aimed at obtaining a model to predict the academic performance of new students based on some socio-demographic and academic variables. The predictive attributes upon which the model was built relied on first semester records of students at a School of Engineering. The performance of students were classified as low, middle, or high depending on the number of course passed or failed in first semester using the Naïve Bayes classifier in the Rapidminer software environment; a model of high accuracy was reported.

Similarly, the prediction model constructed in (Sharabiani et al., 2014) to forecast the academic performance of the Engineering students was also based on the Bayesian networks framework. The study with the specific objective of predicting the students' grades in three major second semester courses explored the repository of the undergraduate engineering students at the University of Illinois in Chicago. Upon testing the developed model against some of the similar models reported in the literature, the study reported to have outperformed them in grade prediction.

The proposed study reported in (Mohammad & Almahmeed, 1988), was aimed at examining the usefulness of the traditional admission standards used by Kuwait University for the prediction of students' academic performance. Findings from the study revealed that, students' secondary school scores reflect intellective as well as the non-intellective factors pertaining to students' background. This study therefore, justifies the inclusion of scores in high school, as one of the attributes being considered as input data in the creation of predictive models in the present research.

The use of fuzzy logic approach to create an expert system for the modelling of academic performance of students was reported in (Yadav & Singh, 2011). Although, the use of soft technique is good for knowledge representation but the approach appears not to be suitable when dealing with very large dataset. Besides, it does not have the capability to learn, therefore, it cannot generalize. The learning capability of neuro-fuzzy approach makes it more robust and a preferable technique to fuzzy logic.

The study reported in (Do & Chen, 2013) used the approach of neuro-fuzzy for the classification of students' academic performance. The use of this approach involves the hybridization of the learning capability of the neural network with the knowledge representation technique. In the study, the performances of students were grouped into poor, average and good. Cross validation was used to determine the level of accuracy of the model classifiers. A confusion matrix describes those accurately classified output data and those that were wrongly classified. It was on this basis that, the level of accuracy was determined. With the use of this technique, there is possibility of rule generation which adds to the understanding of the process.

The use of fuzzy probabilistic neural network for students' academic performance prediction proposed in (Arora & Saini, 2013) does not require any learning rule to train the network and no predefined convergence criteria needed to be specified. It uses radial basis functions as activation functions in the hidden layer and the approach was reported to lead to a high accuracy of classifying the students' performance.

Through the questionnaire administered at three Belgian universities, the data collected in the study reported in (Vandamme et al., 2007) were modelled using the techniques of neural network, decision tree and discriminant analysis. The study was to predict the group of performance a student belong at the beginning of the academic year in order to have a proper planning for their successful studies. Findings from the study show that it was able to group the 533 students based on their risk status. However, information collected through an instrument such as questionnaire is likely to be very subjective.

There are some limitations that is peculiar to the use of machine learning techniques for the building of models from students' data. The model developed through this technique is usually not interactive enough and it is restrictive. It is restrictive in the sense that, it does only what it is programmed to do which sometimes may not cover what the user wants. Interesting areas of research include how to interact with a data mining procedure, how to incorporate a user's background knowledge in mining, and how to visualize and comprehend data mining results (Rotshtein & Rakytyanska, 2012).

The reviewed studies show the several methods that have been proposed for the creation of predictive models. Efforts were made to achieve accurate predictive models. Some of the proposed studies mainly used machine learning, probability theory, statistical and set theory. Neural networks have a strong predictive capability and it is widely reported for prediction purposes in the literature. It has the capability to model the predictors as a nonlinear combination of the inputs. But it sometimes face the problem of over-fitting; this implies that, despite giving accurate result with the trained data, it sometimes gives inaccurate results when simulated with a set of untrained data.

This research, therefore, focuses on alleviating the problem of over-fitting in feed-forward neural network, which is usually responsible for its inaccuracies with a set of untrained data. Apart from the problem of over-fitting, the reviewed studies have shown other gaps, especially the students' data that were explored for the performance prediction of the fresh students. The students' continuous assessment scores within a single semester, appear not adequate enough to predict or anticipate their future performance. The historical data that comprised of both academic and demographic is explored for a similar set of newly enrolled students in this research.

Also, several studies reported in the literature collected data through instruments such as questionnaire, which most times, may be full of missing values and usually subjective. At times, the mood of those that responds to questionnaire determines their response to questions. The data of the newly enrolled students explored in this research were retrieved directly from an institution's student portal. The data were initially supplied by the students while they were seeking for admission. The data are therefore, detailed, well structured and free of any subjectivity; the originality of the data is also preserved.

Since some strengths and weaknesses of the reviewed studies have been identified, the main problem or questions this research needed to answer were formulated. Specifically, the problem of over-fitting while modelling data using the feed-forward network structure is addressed in this thesis.

Due to the limitations in the use of learning algorithm, and the need to have a more efficient means of exploring students' records, a rule-based algorithm is proposed in this research. This is an exploratory technique that relies on a set of some pre-defined rules. Also, in order to predict the performance of fresh undergraduate students in the present research, their academic antecedents were modelled using the proposed approaches. Useful knowledge were unveiled from their historical data.

CHAPTER 3

AN ENHANCED FEED-FORWARD NEURAL NETWORKS

3.1 Introduction

Two approaches are proposed in order to achieve the aim of this research. The focus of this chapter is on enhancing one of the neural network architecture, Feed-forward Neural Networks (FNN). In this chapter, an algorithm is proposed for better partitioning of the data to be trained. Network models were created using both the existing and the enhanced feed-forward neural networks. This chapter also discusses the design of the network structure, experimentations and validation of the network models created. The method of data collection and how the data were pre-processed into a suitable format for mining, is also discussed in this chapter. Giving some untrained predictive attributes, the main objective of this chapter was to achieve predicted outputs that give almost or the same target outputs. In other word, to create a network predictive model that has a negligible error.

3.2 The Proposed Method

The algorithm proposed in this chapter focuses on creating a predictive network model that improves the generalization of a network model created using the feed-forward network structure. In this approach, the Matlab function that is responsible for the conventional way of data partitioning in Feed-forward Neural Network (FNN) is modified to bring about improvement. The use of default partitioning of data for training purposes is not addressing the problem of over-fitting. There is over-fitting when a model gives very little error during training, but the error becomes large when such model is evaluated with sets of untrained data.

The modification of the function paves way for flexibility as against the use of a fixed partition of data for training. With this approach, the data apportioned for the training process, can be chosen based on the size of the available data from which a model is to be built. This approach allows more predictors within the input space to be trained.

Partitioning of the data for training usually involves dividing the entire dataset into two parts, one part for the training and another part for testing. But recently, as a way of improving training of data, the use of the early stopping approach is becoming popular. This approach is what is used in this research, whereby, data are partitioned into three parts i.e training, validation and testing. This approach provides a validation data that monitors the training process, it converges the training when the objective function begins to rise.

In order to consolidate this approach of reducing over-fitting, this research further explore the area of data division for training purposes using feed-forward neural network architecture. With the proposed enhancement, there is more data for training within the input space, as the data division function can easily be varied.

Also, the proposed method follows a number of steps, first among them is the collection of the data that is used for the implementation of the algorithms. The data collected is pre-processed in order to ensure that they are in a suitable format for mining.

3.3 The Design of the Proposed Methods

The flowchart for the methods proposed in this research is represented in Figure 3.1. The first part of the chart is the collection of data and pre-processing of these data to make them suitable for mining. Pre-processing tasks sometimes encompass attribute selection, transformation and resolving the problems of missing values. This is discussed further in the next section, specifically, the method of data collection is discussed in Section 3.4. Only the excerpts of the data to be used for implementation is pre-processed; the rationales for selecting certain attributes, while some other attributes were considered to have less predictive relevance is discussed in Section 3.5.

Next on the chart is the collection of the techniques used in this research. The multilayer perceptron network is another name for feed-forward neural network structure. Network models are created using both the existing feed-forward neural network and the proposed enhanced feed-forward neural network. The training performance of the models created using both approaches is compared, this was to establish the impact of the enhancement introduced. Also, an evaluation is conducted using a new set of untrained data for all the models created through both approaches. The training performance of each model and the results of evaluating all the models created are represented and discussed in extensively in Chapter 5.

The second method, Rule-based Algorithm, proposed for exploring the students' data in this research is also represented in the chart. The proposed RBA algorithm is implemented using PHP and the code for its implementation is listed in Appendix C. The effectiveness of the algorithm is evaluated by measuring the Accuracy of the outputs that emanated as a result of implementing this algorithm on some students' data. The accuracy of the outputs generated is measured by expressing it in terms of error. The computation of errors and further discussion on implementation of the RBA is shown in Section 4.4.

The EFNN is the new structure that emanated as a result of the enhancement proposed and implemented on the existing FNN. The models created using both the existing and the enhanced technique are simulated using a new set of untrained data. The results are discussed and subsequently summarized in a table. This procedure is repeated when another set of data is used for the implementation. Details on this are provided in Chapter 5.

Also, as shown in the chart in Figure 3.1, the effectiveness of some existing techniques that are reported to have been implemented on students' data for prediction purposes such as Decision tree, Fuzzy logic and Artificial Neural Networks (ANNs) are compared to the proposed RBA. The comparisons made were based on three criteria, these include: Generalization, Accuracy and Transparent process. This research relies on what has been reported regarding these existing techniques in the literature. Especially, as relates to using these techniques for the creation of models that predicts the students' academic performance. Each of these techniques has been reviewed in Chapter 2; also, the discussion of these state-of-the-art techniques reveals their strengths and limitations, this form the criteria on which the comparisons in this thesis were based. The results of implementing the proposed RBA and the findings that were unveiled from the comparisons made with some existing predictive modelling techniques are discussed and illustrated at the end of chapter 5.

The techniques used and and other tasks performed prior to creation of the models and their subsequent implementations is represented in the next chart. The entire procedures or flow chart of the proposed approaches are represented in Figure 3.1. This summarizes the proposed methodology used in this thesis.

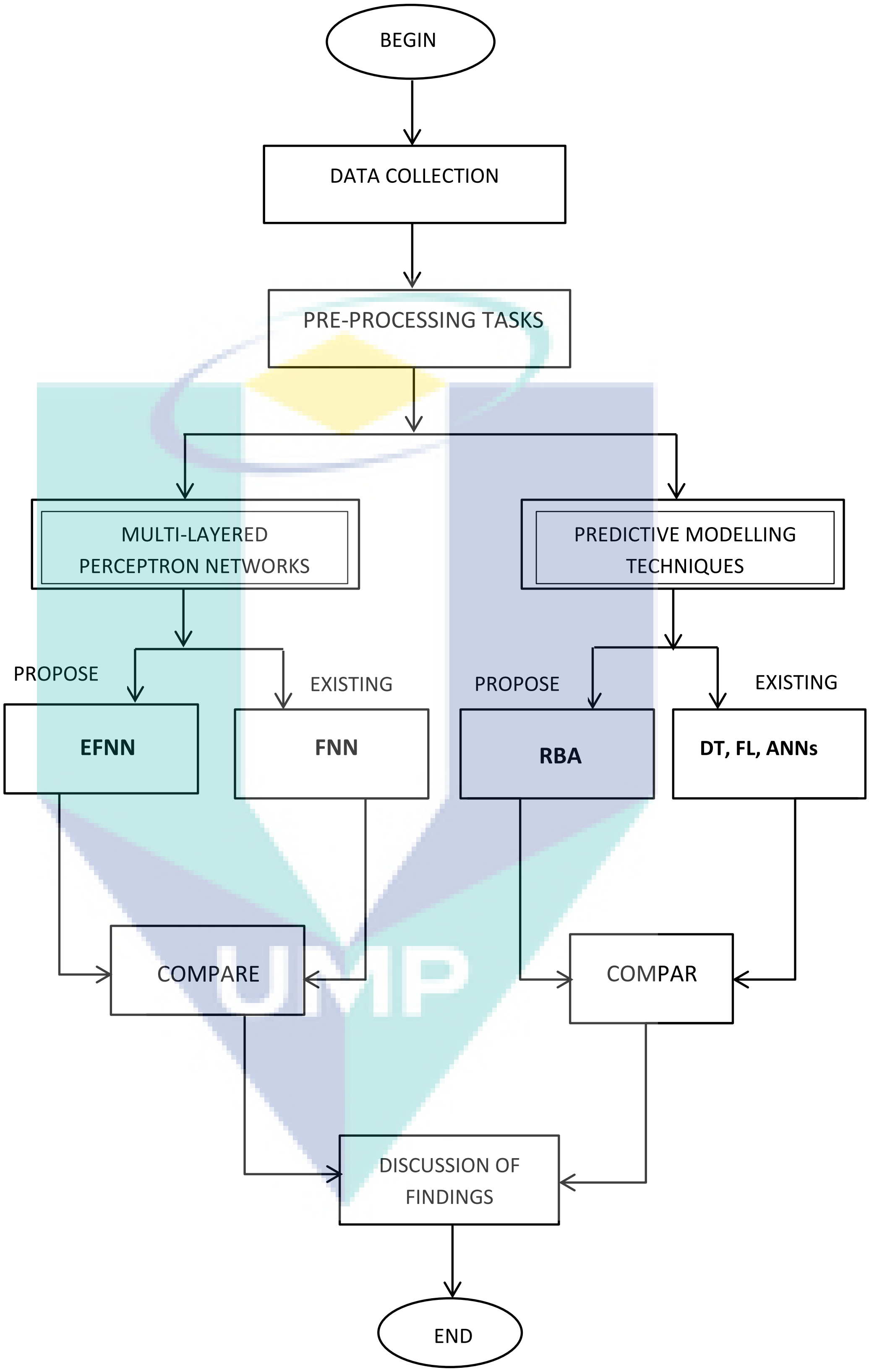

Figure 3.1. The proposed research methodology.

3.4 Data Collection and Preparation

In order to implement the proposed approach, there was the need to get some students' data of the newly enrolled undergraduate students. Also, the data required for this research must be a real-life data and must not have been tampered with or compromised in any form. A public university was contacted in the north central Nigeria, and the records of students that were recently offered admission was obtained. The data collected covers admission offered to students for two academic sessions.

The data originally stored as MySQL file was pre-processed. One of the pre-processing tasks that is inevitable in a data mining research is transformation. There is always the need to transform alphabetic data to numbers, this is because most of the exploratory or learning algorithms can only be implemented on numeric data. For the purpose of easy referencing of these data, and to be able to use relevant commands for transformation purposes, the file is converted to excel file. All the predictors identified in the dataset were transformed into a range of numeric values.

The data set obtained has several variables, but not all the variables have predictive relevance. For instance, variables such as: students' address, phone number, religion, name, and so on have no predictive relevance. Therefore, only those attributes that have predictive relevance were selected and transformed. The selection of predictive attributes is based on expert knowledge and what has been reported in the literature, especially, studies that are related to the proposed research. Initially, 1400 out of the data collected were pre-processed and implemented. To further establish the consistency and accuracy of the proposed algorithms, another set of 2500 data that belong to students of other faculties were implemented, adding up to a total of 3900 students' data.

3.5 The Rationales for the Attribute Selection

The selection of attributes used for prediction is sometimes a big task, as several attributes in the data being explored may not have predictive relevance. A number of characteristics may influence the academic achievement of students and this usually affects the choice of predictors as input data. There are a number of ways of selecting the predictors. However, it has been revealed in the literature that, the best way to select relevant attributes is manually; the selection supposed to be based on a deep understanding of the learning problems at hand and the real meaning of these attributes (Witten et al., 2011).

Predictors would be one-sided if it tied to only academic qualifications. In the present research, the attributes identified as predictors from these records and how best to arrive at the target attribute were based on the information received from the domain expert and related studies in the literature. As reported in (Suh, 2012), attributes of high predictive influence can be suggested by the domain experts. In the light of this, information from experts in the field of data mining were found useful and guided the selection of the predictive attributes that were modelled for students' academic performance in this research.

3.6 The Input Data

A number of words and phrases are used interchangeably in this thesis as input data, these are predictors, input attributes, predictive attributes and input variables. This research adopt the technique of supervised learning method, whereby, a target attribute is provided for the purpose of training the model. The data sets used for creating the network model consist of seven attributes, which is here referred to as *input attributes* and the collective attributes in percentage gives the student's achievement score, which is otherwise referred to as the *target attribute*.

The transformed data has no missing values, the records were properly captured. This happens as a result of the interface that was designed to capture the data sets. The interface had been programmed to validate the correctness of data to be submitted by students. The major pre-processing task in this research, therefore, was to select attributes of predictive relevance. Also, to ensure that data are properly transformed and in a clean or suitable format that can be processed by the learning or proposed RBA.

The predictors identified from the students' records are discussed in the next paragraphs. The seven attributes are presented in Table 3.1. The score achieved by each student as contained in the obtained records were normalized to a range of values between 1 and 6. For instance, if two students score 200 and 220 in a national examination, these scores were normalized, such that, the student has scored 1 and 3 respectively. Table 3.1 provides the details of the normalized values. The description of the attributes that were modelled for prediction is discussed in relation to the environment from where the data were collected. The description of each attribute is as follow:

***SNE*:** This denotes the Score in National Examination. The mode of entry where the data was collected is either through direct (by presenting higher institution result) or by participating in the mandatory examination known as utme (unified tertiary matriculation examination). This is in addition to successful completion of high school with minimum acceptable grades. This entry examination must be attempted by all the candidates seeking admission to Nigerian universities. However, while the direct entry students join the students in the second year at the university, utme students start their study from the first year.

***HSR*:** This acronym denotes High School Results. The quality of high school result was determined and coded in this research in such a way that, the lowest grade was coded as 1, while the highest recognised grade was coded as 6. The West African Examination Council (WAEC) grading system was adopted. It is worth to note here that, the Senior School Certificate Examination (SSCE) conducted by WAEC is a mandatory international examination at the time of this study for all the high school students in Nigeria and four other countries in West Africa sub-region. These other countries are: Ghana, Gambia, Sierra Leone and Liberia.

***IST*:** This stands for Institution Screening Test. This is the score obtained by students in the screening conducted by the institution to validate their achievements in SNE. There is possibility to have the national examinations compromised and possibly the exercise might be free of any malpractice. To be sure of the students' performance, the institutions created their own internal screening test to validate whatever a student might score from the national examinations.

***NHSEA*:** This is the number of high school examination attempted. Brilliant students write their examination only once and pass all the subjects, while average students may have to attempt such examination one more time to achieve the required result allowed for further studies. The information supplied by student as regards whether he or she has attempted the examination once or twice would add substantial value to academic predictor attributes.

SA: This denotes the Student's Age. Generally, the ability of the human being to recall or retain what is stored in their brain, or ability to learn new things fast varies with age. In a normal condition, younger people are the most favoured. It is a general norm that elderly people have several responsibilities and other important things to think about than the teenagers.

YLE: This is the Year of Last Examination. The last year of contact with the school also plays an important role whenever student decides to further their studies. Someone who had left school for more than ten years may not find learning so easy compared to someone who has no period of interruption. During the period that such persons were out of school, the curriculum might have been reviewed and upgraded. This is why someone that had left school for a very long time, may not be able to cope very well compared to those that have their studies uninterrupted.

SO: The School Ownership determines who fund a school and by extension, who is responsible for the overall control of the school and its activities. From all indications, students in private school have no reason not to perform better than their counterpart in public schools. The facilities they need can get to them without any long bureaucratic process and students are well monitored since the workers are aware of the fact that, their pay is tied to the smooth running of the school.

Table 3.1
The Predictive Variables and their Normalized Values

Input variables	Normalized values	Obtainable values
Score in National Exam. (SNE)	UTME: 200-219 1 220-229 2 230-239 3 240 and above 4 DIRECT: ND/NCE/A'LEVEL 2 HND 3 BSC 4	1 – 4 2 - 4
High School Result (HSR)	GRADES: A1 6; A2, B2 5; A3, B3 4; C4 3; C5 2; C6 1	1 - 6
Institution's Screening Test (IST)	SCORES (%) 50-59 1 60-65 2 66-69 3 70 and above 4	1- 4
Number of High School Exam Attempted (NHSEA)	1 Attempt 2 2 Attempts 1	1 - 2
Student's Age (SA)	14-20 3 21-29 2 Above 30 1	1- 3
Year of last Exam. (YLE)	1 >= YLE <= 4 3 5 >= YLE <=8 2 YLE > 8 1	1 - 3
School Ownership (SO)	Public 1 Private 2	1 - 2

3.7 The Target Data

The target variable is related to what each individual student achieved after taking into consideration all the predictors listed earlier in Table 3.1. The higher the value achieved from these predictors, the better for the student. The maximum obtainable value is 50 i.e if all the maximum obtainable values in respect of all the predictors are added together. This is further standardized to 100 percent that forms the target; this is computed for each student based on the formular represented in Eq. (3.1), so that, the individual student's achievement can be expressed in percentage. For instance, if a student achieved the following in the input attributes: SNE=3 ; HSR=22, i.e the value achieved from 5 relevant subjects ; NHSEA =1 ; Age=2 ; STI=2 ; YLE=2 ; SO=1, this gives a total of 33, using the formular represented in Eq. (3.1), the student has gotten 33/50 *100 = 66% . By following this procedures, the computation was done for all the students and arranged as inputs, target, in a table format.

$$\frac{y}{Q} \times 100 \qquad (3.1)$$

where y is the obtained value and Q is the total obtainable value.

The value achieved by each student based on the obtainable value shown in Table 3.1, is computed and stored in a new file containing all the predictors and the target variable. The data is arranged such that, the actual output appears last. The target is refered to as the actual output here, while the resulting outputs from the network model is the predicted output.

3.8 Dynamic Partitioning of the Data for Training

Feed-forward Neural Networks, is a powerful network structure capable of modelling the class prediction as a nonlinear combination of the predictor attributes. It has been successfully used for the building of predictive models from educational data as reported in (Arsad et al., 2014). Although, the network is known for fitting accurate model from normalized data, however, there is need to tackle the problem of over-fitting. A network is said to over-fit, when it loses the ability to generalize between similar input-output patterns (Haykin, 2009). After creating a network, the feed-forward neural network structure partitions data for training in Matlab implementation based on the function:

[trainInd,valInd,testInd] = dividerand(Q,trainRatio,valRatio,testRatio).

This function returns data that is partitioned into the ratio 60: 20: 20. The ratio stands for 60% for training set, 20% for validation set and 20% for testing set. Predictive network model created using the feed- forward neural network, usually have their data partitioned based on this function by default.

In the proposed approach, in order to make the division of the data sets to be dynamic, the random partition in the existing network is modified. This was done by assigning the divide function to nil, so that, the algorithm proposed would be used instead of the default function, see Figure 3.2. Also, in step 3 of the same figure, the divide function (divideFcn) is assigned to divide indices (divideInd) to replace the default divide random (dividerand) shown in the function above.

The proposed data division for training makes it possible, to have the data sets division that increases the training portion of the data to be trained in the input space. The data to be trained, the validation data and the data meant for testing the model, were assigned to their respective indices as shown in steps 4, 5 and 6 of Figure 3.2. The network parameters such as the maximum fail, error goal, number of epochs, validation check etc were also assigned to the indices. The network created is trained and the graphs that illustrate the training performances are represented in chapter 5 of this thesis.

Steps 1-8, is the major concern of the dataset to be partitioned for training, while steps 9-11, is to enable proper training of the network in order to display the performance and graphically illustrates the training process. Also, as shown in Figure 3.2, the input section is not actually part of the algorithm steps, but the pre-processed data must be fed as input for subsequent training.

Input:

Normalized dataset

Output:

A trained network model

Method:

1. Network_name = newff (inpt, tagt, n) ; // create network model
2. Initialize the divide function to nil // off the default divide function
3. Network_name.divideFcn ='divideind' // assign divide function to the network created

***** Partitioning of the data for training, validation and testing

4. trainInd = 1: k // *k* is the size of data to be trained
5. valInd = k+1: m // *m* is the size of validation data
6. testInd = m+1: y // y is the size of the data to be partitioned
7. Assign network parameters to indices in 4, 5, and 6

***** Assignment of divide parameter input indices to corresponding input data

8. [traininpt, valinpt, testinp] = *divideind* (inpt, traInd, valInd, testInd)
9. Configure the network // set parameters for network optimum performance
10. Network_name = train (network_name, inpt, tagt) // train the network
11. Show the network training performance

Figure 3.2. Algorithm for dynamic partitioning of the data for training in feed-forward neural network architecture.

3.9 Design of the Network Model

The architecture of multilayer perceptron, feed-forward network is designed and represented in this section. The function used for the data division in this structure is index. With the enhancement proposed, the data sets are partitioned in such a way that, any desirable size of data can be used for training, validation and testing as against the default. The learning algorithm used for training is a Levenberg Marquardt algorithm, popularly referred to as back propagation algorithm. The Mean Square Error (MSE) is the performance or objective function. In the process of training, computation of error is done at the end of each iteration, this make the changes in error to be quickly noticed. It is the responsibility of the validation data to confirm the rise in error value when the end of iteration is reached. The training process converges as soon as increase in error is noticed.

The number of neurons used is 10. Although, neuron normally increases the processing power of the network, however, too much neuron in a network can lead to over-fitting. The error goal is set to zero, because there was no need to make provision for error. In case the network cannot be trained to achieve zero error goal, the maximum fails is set to 10. This implies that, the network should converge only if unable to meet up with other settings after a trial of 10 times. The iteration is set to 800, but if other configurations settings are met within a few number of epochs, the network would still converge.

Table 3.2
The Network Configuration

Algorithm	Data Division: Index Training: Levenberg Marquardt
Properties	Network type: Feed-forward BP Performance: Mean Square Error Number of Neurons: 10
Parameters	Epochs: 800 Goal: 0 Min grad: 1e-7 Max_fail: 10 Mu : 0.0001 Validation checks 6

In the Figure represented in figure 3.3, the first layer on the left is the input layer, where the model inputs enter the mathematical formula. The $x_1 \ldots x_7$ represents the predictive attributes that form the input. The final layer on the far right is the output layer. The layers in the middle are not exposed to the users and are usually referred to as the hidden layers. Each layer consists of a set of nodes, each of which receives value from the previous node layers to the left, performs a mathematical operation on these values, and passes further values to the nodes in the right direction.

The typical operation of each node is to sum the values going into it from the previous nodes, and the validation data checks of the error against the threshold value. Propagation of the node outputs to the right multiplied by a layer weight W continues until the network eventually converges. The convergence occurs when error begins to rise.

The output is usually determined by the activation function or otherwise known as transfer function. The sigmoid activation function is used in all the networks created. It is an increasing function that exhibits a graceful balance between linear and non-linear behaviour (Haykin, 2009). An example of a Sigmoid function is the logistic function defined by:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{3.2}$$

where 'a' is the slope parameter of the sigmoid function. The activation function, denoted by $\varphi(v)$, defines the output of a neuron in terms of the induced local field v.

One of the methods that can significantly improve the performance of the neural network model, is to train it with back-propagation algorithm. This is to normalize the input data to a format that can make it train much better. The normalization of the input data in this research involves computing the mean value of each input attribute of the data shown in Appendix D. The mean value of the input attribute was used to average the whole training sample. This was done for each variable in the input space. The intent was to make the training sample close to zero, otherwise, the values would be small compared to the value of its standard deviation. The merit of normalizing the input is to accelerate the back-propagation learning process. The mean of the input data is computed based on the formula in Eq. (3.3).

Apart from normalizing the input attributes, the parameter settings also play a key role in the course of fitting network model from a set of data. For instance, the choice of hidden neurons determines the processing power of the network. The more the number of hidden neurons, the more the ability of the network to recognise the existing patterns (Negnevitsky, 2011). However, excess use of neuron in the hidden layer can lead to over-fitting.

$$\bar{X} = \frac{1}{n}[x_1 + x_2 + \ldots\ldots + x_n] = \sum_{i=1}^{n} \frac{x_i}{n} \tag{3.3}$$

where $\bar{X}$ = mean, x_i = the values of the input attribute

$\sum$ = total sum of the values

n = number of sample cases

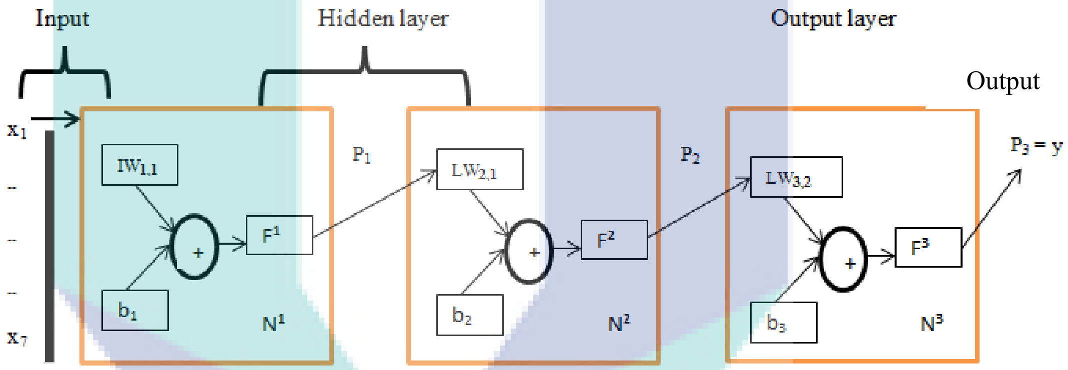

Figure 3.3. The feed-forward neural network architecture.

The learning algorithm for this structure is back-propagation; the reason for its frequent usage is due to its flexibility. Also, it is mathematically tractable. From Figure 3.3, the outputs from each layer can be identified as:

p_1, p_2 and p_3.

$p_1 = f^1(IW^{1,1}\, i + b_1)$; $p_2 = f^2(LW^{2,1}\, p_1 + b_2)$; $p_3 = y = f^3(LW^{3,2} p_2 + b_3)$

combining the three layers would give the predicted outputs: y

$y = f^3(LW^{3,2}\, f^2(LW^{2,1}\, f^1(IW^{1,1}\, i + b_1) + b_2) + b_3)$

N^1, N^2, N^3: number of neurons in each of the layer 1, 2 and 3 respectively

f^1, f^2, f^3: the transfer function for layer 1, 2 and 3 respectively

b_1, b_2, b_3: the bias for layer 1, 2 and 3 respectively

$IW^{1,1}$: input weight matrix connection from input to the 1st layer

$LW^{2,1}$: layer weight matrix connection from 1st to the 2nd layer

$LW^{3,2}$: layer weight matrix connection from 2nd to the 3rd layer

3.10 Implementation of the Proposed Enhanced Algorithm

Three experiments were carried out in the process of implementing the enhancement introduced to the existing feed-forward neural network using the Matlab codes listed in Appendix B. The network architecture model represented in Figure 3.3 illustrates the proposed model. It should be noted that, the proposed network model and the existing network model, shares the same physical features, however, they are differ in the way they partition data for the training of the network model. This is a software issue and not a structure that can be differentiated physically. Therefore, in order to avoid duplication of figures, only the network predictive model represented in Figure 3.3 is used to illustrate both the architecture of the existing and the enhanced network models.

3.10.1 Experimentations

The experiments carried out in this research for creation of network models were implemented in two phases. In each phase, different sizes of data were used for the purpose of training the network. The data in the first phase comprised of 1400 and another set of similar data used for network training in the second phase of the experiment comprised of 2500. In all the experiments carried out in both phases, eight network models were created.

In using the neural network techniques to solve a problem, the right network architecture for a particular task is often chosen by means of heuristics, and designing a neural network topology is still more of Art than Engineering (Negnevitsky, 2011). A typical neural networks architecture comprised mainly of three layers: the input, hidden and output layers.

In this thesis, the predictive network model created is shown in Figure 3.3, and all the settings of the proposed EFNN and the existing FNN models conform to this network structure. Three experiments were conducted in each phase, the third experiment was mainly for evaluation purposes, i.e to determine the error associated with each model created by using set of untrained data. Detail explanation on each phase is as follows:

PHASE 1: The data pre-processed for training in this phase comprised of 1400 data. The data is divided into two parts, a portion comprised of 1200 was used for training the network, while the remaining 200 was used for evaluation. In this phase, three experiments were carried out in which four network models were created and evaluated.

Experiment 1: The network model is created in this experiment based on the existing feed-forward neural network structure. The division of data cannot be varied here i.e the data division for training is not dynamic; therefore, the default partition is used. As earlier mentioned, the FNN by default uses *dividerand function* to partition data for training, validation and testing purposes. With this approach, a sample size of 1200 data was trained in this experiment, using MSE as error function. The training used the concept of a supervised learning, as each input is associated with a specific desired target pattern.

During this experiment, the weights are synthesized gradually, and each step of the learning process are updated so that, the error between the network's output and a corresponding desired target is reduced. The learning by the network continues as the weights of the network are incrementally adjusted. The validation data converged the training process as soon as the error begins to rise. The performance of the training, validation and testing data were then graphically illustrated as shown in chapter 5 (see Figure 5.1). The graph also shows the error values and the number of epochs at the time of convergence.

Experiment II. The procedures that were followed for creating the predictive model in this experiment are similar to the process discussed in *Experiment 1*. However, the procedure followed in the partitioning of the data that is used for training in the present experiment conforms to the proposed algorithm shown earlier in Figure 3.2. With the

implementation of this algorithm, the division of data for training becomes dynamic i.e it varies and no longer static. For instance, the first network model created in this experiment follows the following partition: 64:18:18 ; As indicated in the proposed algorithm, steps 4, 5 and 6 of the algorithm direct the network to partition and assign the partitioned data as follows during the training process: trainInd = 1 : 768 ;
valInd = 769 : 984 ; testInd = 985 : 1200.

These statements are meant to instruct on how the 1200 data required for training should be partitioned. While 768 of the data is used for training, the next 216 was used for validation during the training process, while the remaining data were used to test the model. After the network model is created, there was the need to search for more partitions to create more models. The next partition used for the creation of the third network model is 74 : 13 : 13. Just like the previous partition, the size of the trained data is:
trainInd = 1 : 888 ; valInd = 889 : 1044 ; testInd = 1045 : 1200.

It should be noted that, after the division of the data for training, a number of assignments still follows as shown in the algorithm, this is then followed by parameter settings and the actual training of the network. The detail code is represented in Appendix B. The last network model is created based on the partition 80 : 10 : 10 and the selection of the data for training conforms to these statements: trainInd = 1 : 960 ; valInd = 961 : 1080 ; testInd = 1081 : 1200.

This is how the four network models were created and trained. The graph illustrating the training performance of each network also revealed the best validation perform and the number of epochs at the time. This is otherwise refered to the iteration at which the training converges. The error that occurred during the training process is calculated as the difference between the target output and the network output.

As an effort to reduce the average of the sum of these errors, the back-propagation algorithm adjusts the weights and biases of the network. The method used in dividing the data in this experiment, thus, allows flexibility and to a large extent, alleviate the problem of over-fitting. Another experiment is carried out to evaluate all the four network models created. This is to ascertain the error associated with each of the model and for the purpose of knowing the effect of making the partitioning of data for training dynamic.

Experiment III. The motive of performing the third experiment was to evaluate all the four network models created in the Experiments I and II through simulation. This is important in order to know the level of accuracy of these models, and in particular, the effectiveness of the enhancement introduced in Experiment II. Simulating a network model requires a new set of data that has not been trained. Unlike the two previous experiments, the target data is not required in the process of simulation; only the input attributes are required.

Also, in the previous two experiments, out of a total of 1400 students' data, the network predictive models were built from 1200 students' data. The leftover of 200 data was then used to simulate the four models created in the previous experiments. When a model is simulated, it generates some network outputs, this output and the target is what is required to compute the error associated with that network model. The resulting outputs of simulating all the network models that were created from the two experiments is presented in chapter 5 of this thesis.

PHASE II. In order to establish the consistencies of the proposed algorithm for enhancement, 2500 similar data that belong to students of another faculties were pre-processed and used to implement the proposed approach. Four more network models were created and subsequently simulated as done in the previous experiments.

***Experiment* 1:** Out of the available 2500 data, only 2000 was used for training, the left over of 500 data was reserved for evaluation. The first network model created conforms to the existing structure of FNN, i.e a data division approach that partition data to be trained into the ratio 60:20:20. All the necessary parameters were set and a network model is created. The training performance regarding the errors, number of epochs at the point of convergence and the best validation performance are represented and discussed in chapter 5.

***Experiment* II:** The network models created here are based on the algorithm represented in Figure 3.2. Network models are created based on three different partitions, just as discussed earlier in phase 1. The motive was to try other partitions outside the default represented in experiment 1, this is with a view to achieving a network model of better accuracy. As shown in the algorithm proposed, steps 4, 5 and 6 of the algorithm handles the fractional part of the data to be used for training, validation and testing purposes. For instance, the first network model created here follows this partition: 66 : 74 : 78 and in the process of searching for other partitions that can give accurate models, 74 : 13 : 13 and 78 : 11 : 11 were tried.

The selection of data by the algorithm represented in Figure 3.2 for training based on this partition 66 : 74 : 78 is as follows: trainInd = 1 : 1320 ; valInd = 1321 : 1660 ; testInd = 1361 : 2000. This is followed by assignment of functions, including settings of parameter, thus, the second network model is created.

In creating the third network model in this experiment, the partition 74 : 13 : 13 was used. The selection of the data for training follows the same process as in the creation of the second network model. The first set of 1480 data, out of a total of the 2000 data is used for training, while 260 of the data is used for validation and testing respectively. This is specified in the algorithm as: trainInd = 1 : 1480 ; valInd = 1481 : 1740 ; testInd = 1741 : 2000. After configuring the network, and making all the necessary settings, the third network model is created.

The fourth network model is created by exploring one more partition, 78 : 11 : 11. With this partition, the validation and testing data have reduced to 11. This partition trains 1560 data out of the total 2000 data, validation takes 220 while the remaining portion is used for testing. The training process went well like the previous models created, the graphical representation of the whole process is generated and they are all illustrated and further discussed in chapter 5.

***Experiment* III:** The purpose of this experiment is to evaluate all the network models created in this phase. This is to determine the effectiveness of the models created with a set of untrained data. In the process of evaluating the models created, 500 untrained data were used to simulate all the four network models created in this phase. The evaluation process is further discussed in the next section. The comparison was then made based on the error associated with each of the models created.

The two sets of the data implemented in the experiments carried out, successfully created eight network models. The first model created in each phase is based on the existing FNN structure, while the three other network models that were created through experiment II of each phase are based on the proposed approach, EFNN. To avoid duplication of figures, only one feed-forward neural network architecture shown in Figure 3.3 is used to succinctly represent all the structures of the eight models created. This is because, they all have the same physical appearance, but only differs in the function that control the way each of them partition data for training. This is strictly a software issue; however, the eight graphs that illustrate their training performance are generated and represented in chapter 5.

The training processes during the experiment regarding the update of weight, bias and error computations are explained further for clarity sake. In all the experiments carried out, training was done using back propagation algorithm.

3.11 Learning Algorithm and Training Process

The algorithm used for the training of data in this research is Backpropagation. Although, many neural network models have been proposed, the backpropagation is the most widely used model in terms of practical applications; this assertion does not based on statistical surveys but perhaps, over 90% of commercial and industrial applications of neural networks use backpropagation or its derivatives (Munakata, 2008).

Learning in a neural network is normally accomplished through an adaptive procedure, that is usually referred to as learning rule or algorithm. The process of learning can be viewed as an optimization process, this is because, the activities involve a thorough search in a multidimensional parameter space for a solution. This is to achieve a result that gradually optimizes a pre-specified objective function.

After setting up the network architecture, then the neural network is ready to *learn*, or in another word, the neural network is ready to be trained. Training of data using this algorithm usually follows a number of steps. The actual training takes place within the hidden layer. For each pattern, these three steps should be repeated until the neural network can consecutively map all patterns correctly. Assuming **y** is the output vector, it follows that, the training should be allowed to continue until the network output is equal or close enough to the target vector **t** for the given input vector **x**.

Step 1. Input ***x*** to the neural network.

Step 2. Feedforward. The direction of moving through the neural network, goes in the direction from the input to the hidden layers, then from the hidden to output layers, and the output vector ***y*** is produced.

Step 3. Backward propagation of error corrections. The value of **y** is compared with **t**. This comparison is done by the validation data. If **y** is equal or close enough to **t**, then, the movement is directed to the beginning of the output layer. But in case it is otherwise, then backpropagation occurs through the neural network and adjust the weights so that, the next **y** is closer to **t**, then it goes back to the beginning of the hidden layer. Apart from Backpropagation algorithm, other common training algorithms are: Bayesian Regularization, Scaled Conjugate Gradient, One Step Secant etc.

Epoch: In the above training process, each output layer iteration is called an epoch. An epoch is one cycle through the entire set of patterns under consideration. By taking the first layer as I and the second as J; the learning process commences by first initializing the input weights and biases in the network. The input to each of the hidden units is processed and the activation function computes the output. For instance, to compute the net input of layer *j*, the previous layer *I*, must be taken into consideration. Eq. (3.4 – 3.8) are excerpts of the backpropagation algorithm listed in Figure 2.3. as reported in (Rotshtein & Rakytyanska, 2012).

$$I_j = \sum_i w_{ij} O_i + \theta_j \qquad (3.4)$$

where I_j is the net input of the j unit, θ_j is the bias, w_{ij} is the weight connection from *j* unit to the *i* unit, O_i is the output from layer *i*. The output that serves as the input to the next layer continue to propagate forward, as far as the termination condition is yet to be satisfied.

As the error is back propagated since the set threshold is yet to be met, some increments and updates become necessary. The weight increment:

$$\Delta wij = (l)\, Err_j O_i \tag{3.5}$$

and the bias increment:

$$\Delta\ \theta_j = (l)\, Err_j \tag{3.6}$$

Both (3.5) and (3.6) are updated before forward propagation starts again for the next iteration. The weight update is based on Eq. (3.7):

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \tag{3.7}$$

while the bias update is based on Eq. (3.8):

$$\theta_j^{new} = \theta_j^{old} + \Delta\ \theta_j \tag{3.8}$$

The training algorithm updates the network's weights and biases in the direction in which the performance function decreases most rapidly. One iteration that follow this procedures as shown in (Beale et al., 2010) is represented as Eq. (3.9).

$$x_{k+1} = x_k - \alpha_k g_k \tag{3.9}$$

where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate. This equation is iterated until the network converges.

This is how the training process goes within the hidden layers, say i and j. The output layer receives the final processed outputs when the training process converges.

3.12 Error Computations

In order to know the level of accuracy of the models as the training progresses, the error associated with each model is computed using the performance measure, MSE. The MSE was computed based on Eq. (3.10), and monitored by the validation data. It is the responsibility of the validation sets to keep track of the error function as the training progresses, and to signal the end of training by prompt convergence of the training process when an increased in error is noticed. The error is usually computed with respect to the next higher layer. Objective functions perform important roles in the course of training data and as reported in (Wu & Coggeshall, 2012), the mean square error represented in Eq. (3.10) is one of the commonly used function.

$$\text{MSE} = \frac{1}{Q}\sum_{k=1}^{Q} e(k)^2 = \frac{1}{Q}\sum_{k=1}^{Q}(t(k) - a(k))^2 \qquad (3.10)$$

The MSE is the mean $\frac{1}{Q}\sum_{k=1}^{Q}$ of the square of the errors $(t(k)-a(k))^2$ where Q is the size of the observation data, t(k) is the target value; a(k) is the network predicted value. The attribute that represents the target during training and the network predicted outputs sometimes differ slightly. The square of the differences in the value gives the error which must always be positive, especially when the transfer function used is sigmoid. This evaluation occurs during the training process, it is also necessary to measure the accuracy of the model after training, in order to establish its reliability. The training performance is graphically illustrated and further discussed in chapter 5.

3.13 Evaluation of the Network Models

When all the training process is concluded, it is important to determine the efficiency of the developed models. The two network models created using the existing FNN and the six other network models created based on the proposed algorithm in Figure 3.2 were all evaluated. This was aimed at determining the error associated with each model. The evaluation process involves taking only the input part of the untrained data to simulate the models. This generates some predicted outputs, the outputs generated and the expected outputs are the data required to compute the mean absolute error. The Mean Absolute Error (MAE) was the choice of error measurement here, due to the fact that, it does not tend to exaggerate the presence of outliers and as reported in (Witten et al., 2011), MAE treats all sizes of errors evenly according to their magnitude. The formular used for error computation is represented in Eq. 3.11.

The response of each model at this stage clarifies the actual impact of modifying the partitioning function of the existing FNN. This further shows whether the intent of the enhancement introduced, which was aimed at reducing the problem of over-fitting of the network model is achieved.

Network models were created based on several partitions during the experiments conducted and the accuracy of each model was computed. The computed errors in respect of each partition is recorded in a table and comparisons were made. The detail of the computed errors can be found in Tables 5.1 and 5.2 (see chapter 5).

The error measurement of classifiers such as decision trees, are quite different from this approach. This is because, their target output is usually not numerical, but rather, alphabetic. The present prediction model uses continuous values as a target, hence, the accuracy can only be determined through the measurements discussed earlier such as : MSE, MAE, RME etc. Study proposed in (Willmott & Matsuura, 2005) revealed the advantages of using MAE shown in Eq. (3.11).

$$\text{MAE} = \frac{p_1 - y_1| + \ldots. + |p_n - y_n|}{n} \qquad (3.11)$$

where p is the target output, y is the predicted output and n is the size of the data. It is possible to have the differences in p and y to be negative; this may happen because, the predicted output can either be the same as the target value or greater and sometimes, it may be lower. This is a function of how well the model had learnt; but the Mean Absolute Error of the differences would still be positive.

CHAPTER 4

THE RULE-BASED ALGORITHM

4.1 Introduction

The focus of this chapter is on the second approach proposed for a more efficient exploration of students' data for prediction purposes. The algorithm proposed and implemented is not capable of learning, but it is an exploratory technique that is based on certain pre-defined rules. The prediction of academic performance of the newly enrolled students using this technique relies on some centralized range of scores previously achieved by the students. This achievement (score) computed based on their past data determines how their performance can be predicted. Also discussed in this chapter, is how the user can interact with a system that is developed through the implementation of the proposed approach, this is illustrated using a sequence diagram.

4.2 Using the Proposed Rule-Based Algorithm for Prediction

The technique proposed and implemented in the previous chapter has been able to achieve a network predictive model that gives a much better accuracy, especially when evaluated with a set of untrained data. However, there are several other useful and specific information that the data analyst or institution's administrators may want to know about the students' academic performance. This may be necessary for the purpose of planning; such information may include: summarized or aggregate information about the entire predicted outputs, information on a particular set of students in a particular order and within a particular range of numbers, etc. Instances like this require an algorithm that can efficiently analyse the students' data in order to unveil the desired information. In view of this, for proper exploration of students' data, a Rule-Based Algorithm, RBA, is proposed and implemented in this research.

The proposed RBA does not search for patterns in the input attributes for training purposes in relation to the target as discussed in the previous chapter, but instead, maps the input data to a class label based on certain pre-defined rules as contained in the algorithm. The algorithm is designed in such a way that, the students' data are better analysed for prediction purposes.

The rule that needs to be followed in order to map the students' achievement to the class label needs proper scrutiny. In an instance like this, the knowledge of domain expert is usually inevitable. The information sourced from the expert domain in the course of this research was relied upon and helps to arrive at the information in Table 4.1.

In practice, to effectively use the technique of data mining to make an acceptable decision, a cooperative effort of humans and computers is very crucial. Best decisions are only achievable by balancing the knowledge of human experts in describing problems and goals with the search capabilities of the computer (Kantardzic, 2011).

The proposed approach was designed to be simple to adapt as only predictors and students identification number are the required input attributes. The first set of the explored data is represented in Appendix D. It can be seen that, the target is not provided as obtainable in supervised learning, therefore, inference of the explored data were based on Table 4.1.

Table 4.1
Students’ Performance and their Associated Risk Status

Achievement	Performance	Status
70 >= s <=100	Excellent	Risk free
60 >= s <= 69.99	Good	Low risk
50 >= s <=59.99	Average	High risk
0 >= s <=49.99	Poor	Very high risk

where‘s’ is the overall students’ score (achievement) which represents the target output and required for making the inferences.

4.3 The Design of the Algorithm

The algorithm is designed for optimal performance. It is segmented into three main sections, input, method and output section. It is scalable and follow a transparent process in transforming the input data to outputs.

4.3.1 The Input Design

The seven predictors identified in the records forms the input attributes; each occupies a field and they were transformed to numeric values. A field is also provided for the student’s identification number, the field can take alphanumeric data. No target variable is provided here, since this approach does not use a supervised learning technique as observed in the previous approach. The table that contains the input attributes to be uploaded for processing has the format (student_id, inputs). Where the student_id correspond to the unique identification number of each student, the inputs are the seven predictors: score in national examination, high school result, institution’s screening test, number of high school exam attempted, student’s age, year of last exam, ownership of high school attended.

Since the algorithm takes data as input, there is need to properly structure the database and the table file. Twenty fields were created in the table structure, although, only seven were used at the time of implementing the proposed algorithm. The fields are provided in excess, in case, there is need to have predictors of more than seven. This makes the algorithm to be suitable for any data that can be arranged in the form of student_id, inputs. The computations of data and its proper mapping to other variables, in order to make inference within the framework of the rule were implicitly handled.

For the purpose of storing the results of computations made, the algorithm is designed to allow connection to the database and access the table file. The implementation of the proposed algorithm establishes perfect connection between the codes and MySql. The data to be explored is imported into the database and processed with the relevant codes in order to predict the students' academic performance through inference.

Connection to the database is very crucial to efficient exploration of the normalized data. The complete code that implements the connection to the database in this thesis is represented as follows:

Step 1: $host = "localhost"; $user = "root"; $password = ""; $dbname = "student_performance";

Step 2: $con = @mysql_connect($host, $user, $password) or die("can't connect to the database, sorry for any inconvenience!");

Step 3: @mysql_select_db($dbname) or die("Database not found!");

The first step is the assignment of the localhost to the host, being the location where this model is launched. The proposed approach restricts the access by the user for security reason. The accessing the model within the localhost requires supplying "root" as the user and empty space as the password, the model is named as students' performance.

The second step shows the actual connection to the MySQL, if the correct information regarding the username and password are supplied, then, connection would be established. In case, a wrong access information is supplied, a message of inability to connect would be prompted.

The third step shows the connection to a database within the MySQL. If the named database cannot be found, “database not found!” would be prompted. It is the successful connection to the database that can ensure access to the table files.

The next key task in the proposed rule-based algorithm shown in Figure 3.4 is the upload of data. After a database is successfully created and the required table structure is created within the database, the next is to upload the data into the table within the database.

4.3.2 The Methods

After the algorithm is designed to take some sets of data as input, there is need to come up with the methods that would handle the necessary processing. A number of computations is inevitable ar this stage; sortings and several comparisons must be made, results must be returned to the calling function etc. In order to achieve this, the proposed RBA composed of a number of functions that forms the main steps that performs specific task. Some of these functions are:

Function_compute_the_students’ achievement_score: The achievement scores of all the students must be computed based on the data in Appendix D. This is necessary, because the computed value is required as a parameter by another function. It is also on the basis of this score, that the students’ performance are predicted through inference. The value is also required in determining the risk status in respect of each students’ achievement.

Function_the_students_risk_status (score) : Based on the students’ input data, the score achieved by each student was determined, and also the risk associated with such achievement. The value of the score achieved was passed as a parameter to the function. For instance, if a student achieved a score of 70 and above, it is remarked as ‘free risk’. This implies that, no risk is associated with the score such student has achieved. Also, if the score achieved by a student is less than 50, it is remarked as ‘very high risk’. The risk status associated with a particular performance is not chosen arbitrarily, but forms part of the information that was collected from the domain expert as shown previously in Table 4.1.

Function compute_performance (score) : The score achieved by a student and the risk status associated with such achievement is crucial to infer the predicted performance. The performance of a student may be predicted as excellent, good, average or poor. A performance is predicted as excellent when the score achieved is between 70 and 100. A student's performance is predicted as good when a student obtained a score between the range of 60 and 69.99, while a score from 50 to 59.99 is predicted as average. A score below 50 is predicted as poor and have very high risk attached to it.

4.3.3 The Outputs Design

The implementation of the proposed rule-based algorithm processed the data and generates the outputs results in a tabular form. Although, the algorithm takes numeric values as input, the resulting outputs comprised of both alphabets and numeric values. The algorithm is designed to predict the performance of all the students whose data were received as input through inference. Also, the outputs can be viewed and sorted in a particular order for easy comparison.

The output result is designed to be viewed in part or in full, this depends on what the user wants. Due to space constraints, the predicted outputs of all the students whose data were explored are not shown here, but can be found in Appendix E of this thesis. The outputs reveal the identification number of students, the numerical value of what each student achieved and the risk associated with such achievement. The academic performances of all the students were inferred from these variables.

As shown in the algorithm, there is a need to connect to the table file within the database each time inference is to be made and the outputs is to be listed. The algorithm is designed to be scalable, as procedures to handle a new function can easily be added while the necessary procedures to retrieve the output of the newly added function can also be added accordingly. The proposed rule-based algorithm is represented in Figure 4.1.

Input

transformed students' data

Outputs

students' performance predicted, students' risk status
statisitical summary of students' performance predicted

Method

```
F1:  Compute students' achievement score    // using the transformed data

F2:  Compute the students_risk_status (score)  {     // score is required
          Determine the student's score achieved between the range X and Y
          IF (score>=X and score<=Y)                    // Y>X; 70 >= X<=100
          Let the risk status be free
          ELSEIF (score>=X1 and score<= Y1)     // Y1>X1; 60 >= X1<=69.99
          Let the risk status be Low
          ELSEIF (score>=X2 and score<=Y2)     // Y2>X2; 50 >= X2 <=59.99
          Let the risk status be High
          ELSEIF (score < F )                           // F < 50
          Let the risk status be Very high
          RETURN (risk status);

F3:  Predict the students_performance through inference (score) {
          IF (score>=X and  score<=Y)                 // Y>X; 70 >= X<=99.9
          Let the performance be Excellent
          ELSEIF (score>=X1 and score<=Y1)      // Y1>X1; 60 >= X1<=69.98
          Let the performance be Good
          ELSEIF (score>=X2 and score<=Y2)     // Y2>X2; 50 >= X2 <=59.98
          Let the perforamce be Average
          ELSEIF (score < F )                      // F < 50
          Let the performance be Poor
          RETURN (performance)

F4:  Get the students' risk status        {
     CONNECT to the DB to retrieve the student's id and risk status
          RETURN (risk count);

F5:   Get the student's performance predicted ()    {
     CONNECT to the DB to retrieve and sort the student's id and  performance predicted
          RETURN (performance count);
          GENERATE the outputs
              }}}}
```

Figure 4.1. The proposed rule-based algorithm.

4.4 Implementation of the Proposed Rule-Based Algorithm

The proposed algorithm is implemented in PHP language; the code listing is available in Appendix C. This algorithm explores past students' data and the proposed RBA is implemented on it in such a way that, it is interactive and user-friendly. The user an interaction with the implemented algorithm is illustrated using a sequence diagram represented in Figure 4.2.

After the database has been created along with some required tables, the data to be explored needs to be uploaded. A feedback of successful uploads or otherwise, must also be prompted to the user, instances like this were taken care of by the error handler. The next step is the implementation of the method section of the algorithm. The first task is the computation of what should serve as the target.

Unlike the learning algorithm where the input attributes and the target were supplied together. In the present approach, the achievement computed in respect of each student is released by MySQL on demand. This achievement or students' score is passed as a parameter while predicting the students' performance and the risk associated with such achievement is also computed and stored against the students' identification number.

The students' performance is then predicted for all the students by inferring from the achievement computed and the risk status associated with such achievement. The summarized or aggregated outputs are also produced. The implementation further reveals the best performance students of specific range. The options list are created to enable users choose the task they want to perform.

As shown in the algorithm, after the students' achievement is computed, their performance is predicted and their respective status is determined. Each request made at the interface, contact the backend (Mysql), for the outputs to be generated, since it keeps all the processed data and their results. The entire outputs generated with the proposed RBA when implemented with the first set of data, that was used for training (1200) in the previous chapter is available in Appendix E.

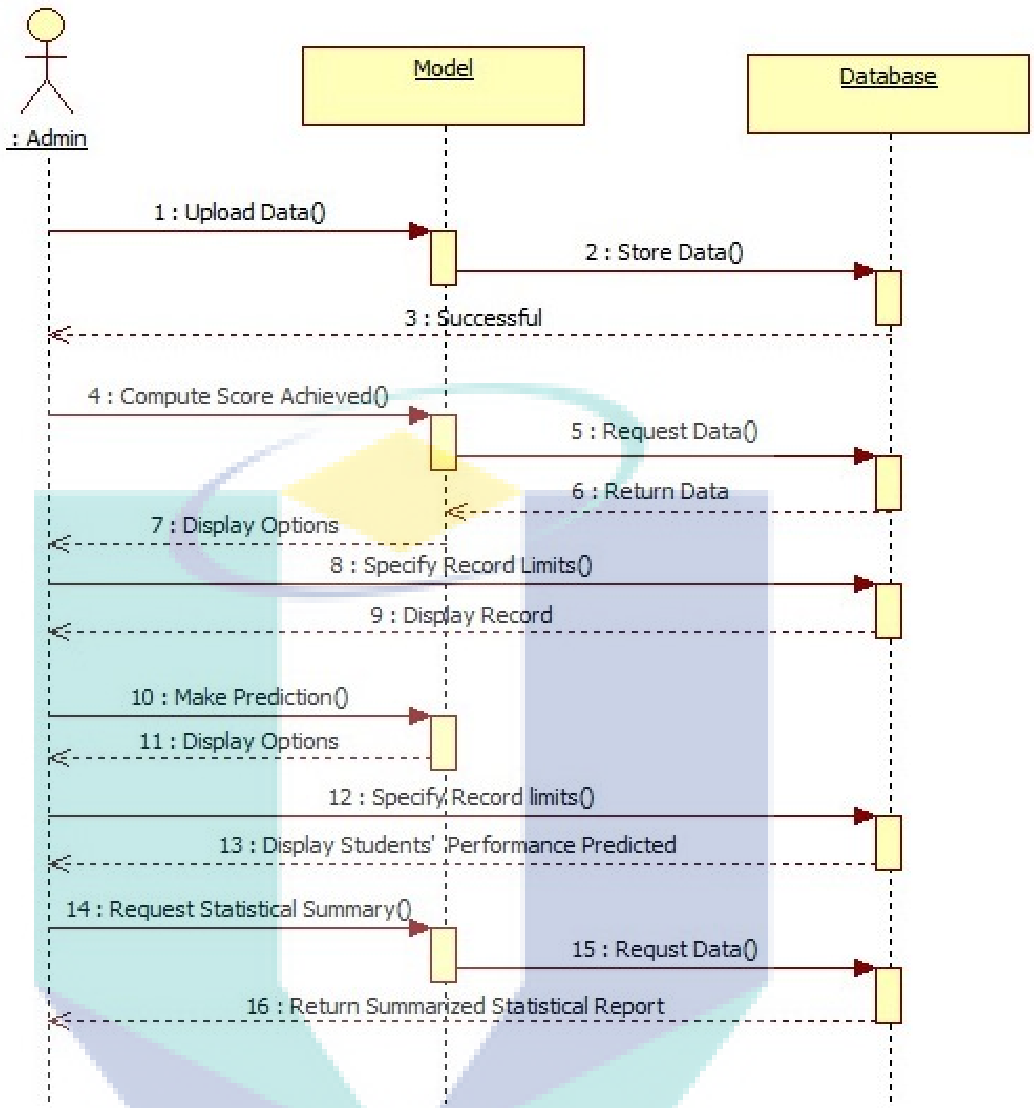

Figure 4.2 . User interactions with the implementation of the proposed RBA.

It should be noted at this juncture that, the concept of data mining is not synonymous to mere querying of database. But querying sometimes complement the procedures used in the process of data mining to achieve desired results. Querying a database can only give the output that is a reflection of the content of what is stored within the database. The algorithm used in data mining is capable of unveiling information that is not directly stored in the database; this is usually achieved through inference. This explains why the data in Appendix D give the outputs represented in Appendix E.

In order to predict the performance of the students using the proposed RBA, this research adopts the knowledge-driven approach. In this approach, the domain expert provides the information in the form of rules. This rule gives a clear statement for easy understanding in the interpretation of the overall scores achieved by each student and how useful inference can be made from them.

Generally, rules consist of two parts: the IF part, which is usually referred to as the antecedent- premise or condition. Also, the THEN part, usually referred to as the consequent- conclusion or action. As reported in (Negnevitsky, 2011), a rule can have multiple antecedents joined by the keywords AND, OR and sometimes a combination of both. The information in Table 3.3 can be expressed inform of statements or rules, that guides in the prediction of students' performance. This was subsequently implemented using PHP. The statement is expressed in the form:

> IF score achieved <50 and the risk status ='Very high risk', THEN performance predicted is '***Poor***'.
>
> IF score achieved >=50 and the score achieved <=59.99 and risk status ='High risk', THEN performance predicted is '**Average**'.
>
> IF score achieved >=60 and the score achieved <70 and risk status ='Low risk', THEN performance predicted is '***Good***'.
>
> IF score achieved >=70 and the score achieved <=100 and risk status ='Risk-free', THEN performance predicted is '**Excellent**'.

The process involved in transforming the input data to become the outputs that are generated followed a clear process of IF statement listed above. The technique allows better user interaction with the model, as this was lacking in the use of learning algorithm.

The resulting outputs of each test that was carried out regardless of the data size were found to be consistently accurate. For instance, a set of another 2500 processed and analysed using the proposed RBA is found to be 100% accurate, when measured using the accuracy metrics. Accuracy can be computed by finding the error associated with the predicted outputs. The percentage error is computed using the formular represented in Eq. (4.1). The most important static characteristics of an instrument is its accuracy, which is generally expressed in terms of the error (Bakshi & Bakshi, 2009). Accuracy is how close a measurement comes to the truth and (Anderson, 2015) gives the error formular as:

$$\% \text{ error} = (AV - YV) \times 100 \div AV \tag{4.1}$$

where AV is the Acceptable Value and YV is Your Value. In the present instance, for the output represented in Appendix E, AV is all the 1200 predicted output that conforms to the rule provided in Table 4.1. The YV in this instance, is the number of each performance, such as good, excellent, poor, etc that correctly matches the expected class label, the addition of this also totals 1200 since none is found to be wrong. Consequently, the computation of the error % gives zero. Similarly, when the proposed RBA is implemented on another set of 2500 students' data, the same zero percent error is recorded based on the formular represented in Eq. (4.1). In order to avoid unnecessary swelling up of this thesis with dataset and the generated outputs, the 2500 data and the outputs it generated is not represented in the Appendix. However, a summarized output that was generated is represented in Table 5.4 (see chapter 5.).

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

This chapter presents and discuss the results from the two approaches proposed for this research. As two different sizes of data are used in the implementation of the proposed approaches, the results that emanated from both approaches are discussed here. Specifically, this chapter discusses the training performance of the network predictive model that were generated during the training process. The error associated with each of the models, and the summarized output generated from implementing the RBA are discussed in this chapter. Some of the key findings of this research are discussed in relation to some other predictive modelling approaches reported in the literature.

5.2 The Resulting Outputs

The results of the two approaches proposed in this research are discussed here. The results of the first approach are divided into two since more than one dataset is used. The outputs from implemening the algorithm proposed for the enhancement of FNN represented in Figure 3.2 on the 1400 dataset is presented as results of evaluating the first set of network predictive models. Similarly, the results of using 2500 datasets are presented here as results of evaluating the second set of network predictive models.

The results of the second proposed approach, that focuses on efficient exploration of students' data for prediction purposes is also represented and discussed in this chapter. It is presented as the outputs results of implementing the proposed RBA. Some results are represented in tabular format, while some are graphically illustrated.

5.2.1 The training performance generated

The Figures 5.1 and 5.2 presents the graphical representations of the training performance of both regular and the enhanced feed-forward neural networks. These graphs illustrate the training performance that is automatically generated to show the trends of the trained data during the training process. The 1200 trained data represents the excerpts of the records of the newly enrolled undergraduate students. The data is normalised and used for the implementation of the network model created using the enhanced feed-forward network. The graphs generated at the end of the training shows the performance of Mean Square Error (MSE) against the number of iterations (epochs).

The training process started on a good note in Figure 5.1 by a continuous reduction in error. In training a data set using a supervised learning approach, there is used to be a threshold. This is a predefined value that must not be exceeded during the training process. The threshold may be in the form of number of iterations or specific error value. Table 3.2 shows a number of parameters, the set values in the table are the threshold values.

As the training continues, the error values also continue to decrease. At the end of each iteration, the value of MSE is being consistently checked by the validation data. The role of the validation data is to terminate the training process whenever the error begins to rise. Figure 5.1 represents the performance of the network predictive model trained with 1200 students' data and its creation is based on the existing structure of the Feed-forward Neural Networks.

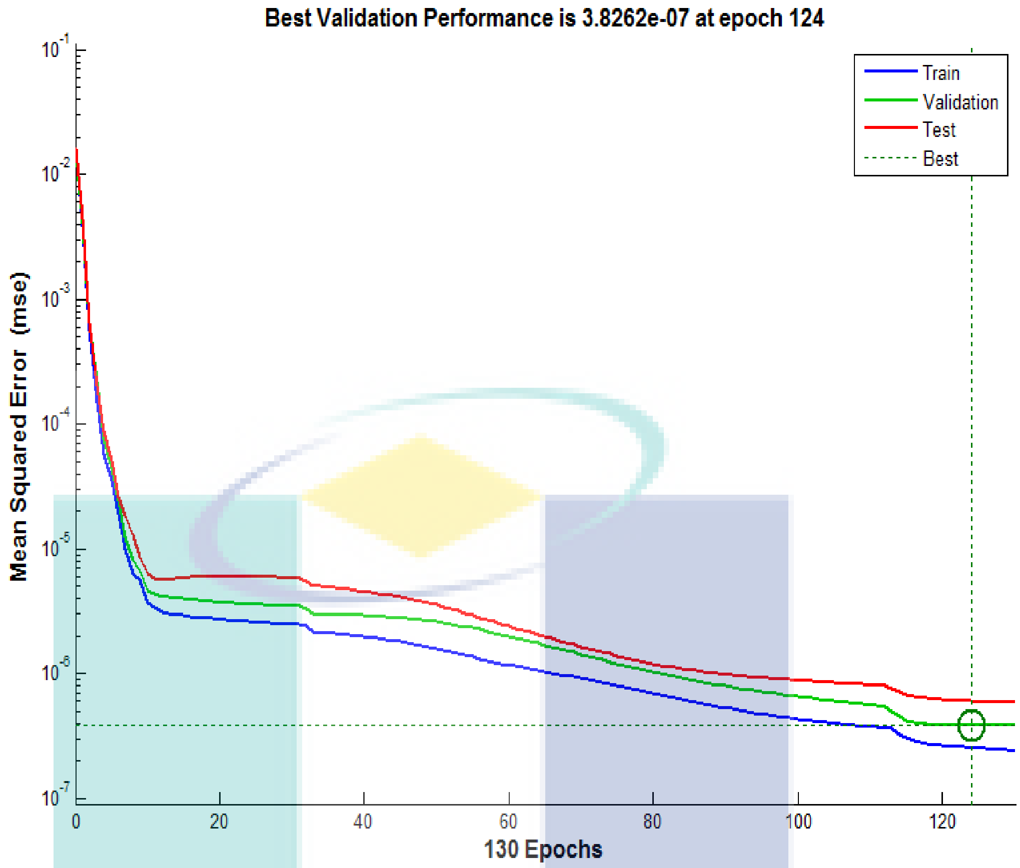

Figure 5.1. The training performance based on regular FNN structure for the first set of data.

In order to shift from the model creation using the existing network technique, the second network model was created by training the same size of data and based on the algorithm proposed and represented in Figure 3.2 (see chapter 3). The algorithm is designed to bring about flexibility in the division of data for training as earlier explained. The word flexibility here implies that, partition of the data for training is not static. A different partition used for training in the ratio 64:18:18 gives the training performance represented in Figure 5.2. It can be seen from the graph that, the training process converges much earlier. Also the best validation performance, i.e. the value of mean square error at the time the validation data converge during the training process is lower compared to the previous partition that is based on the existing structure of FNN, 60:20:20.

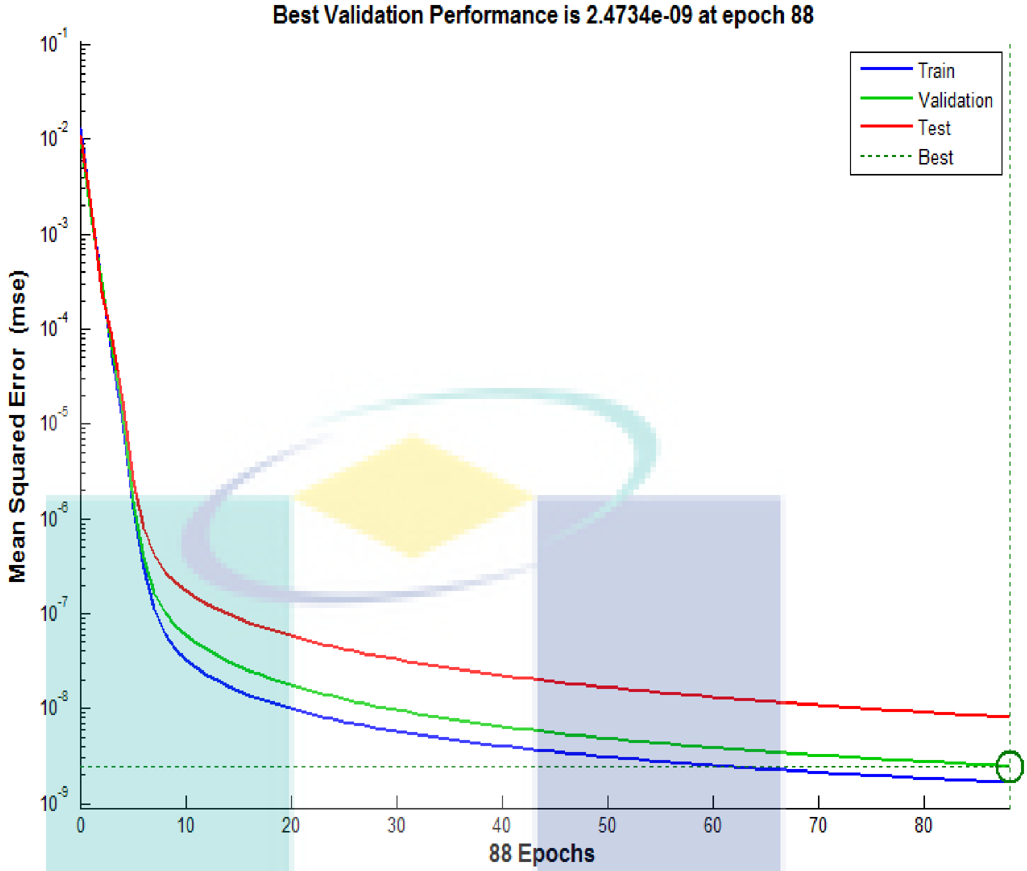

Figure 5.2. The training performance based on EFNN structure (A).

Also, the third network predictive model is based on the proposed algorithm in Figure 3.2. The dataset is partitioned into the ratio 74:13:13. As shown in Figure 5.3, the number of iterations (epochs) reduces compared to the previous training performance, the best validation performance shown on the graph also indicated a much better training process compared to the previous partition 64:18:18. Looking at the graph further, the line that represents the validation and test, exhibit some similarities and at the time of convergence, no sharp increase in error is noticed in the validation data.

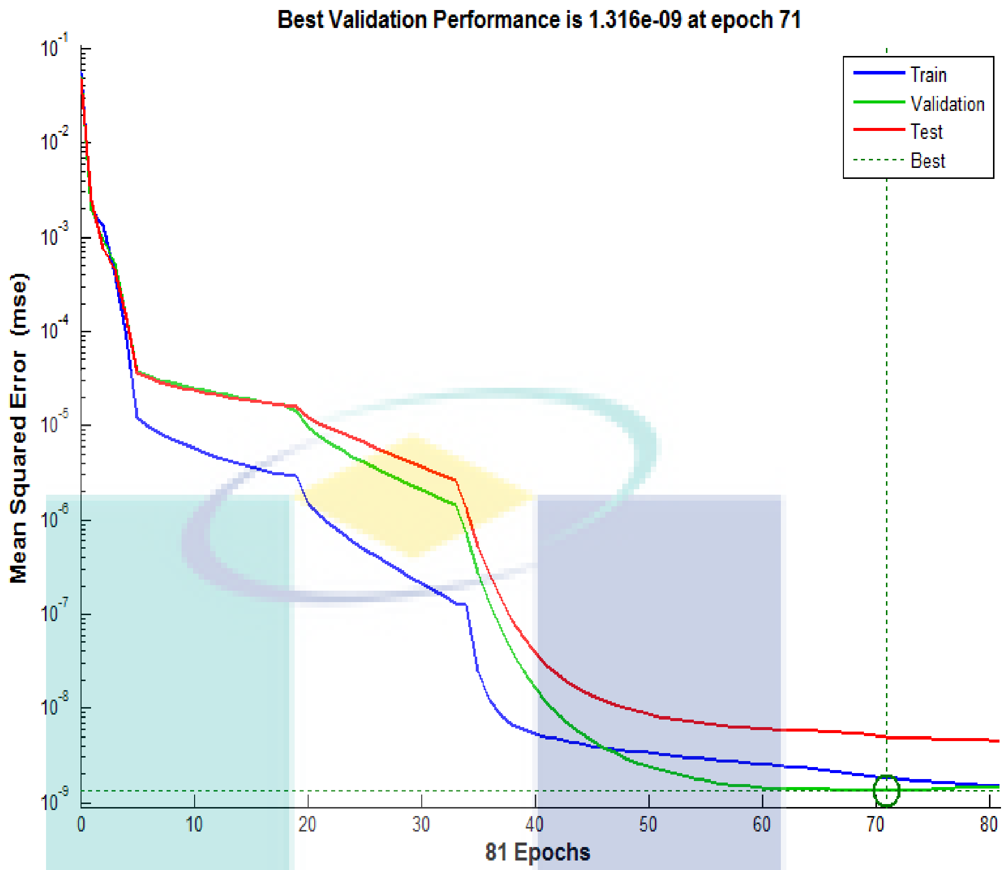

Figure 5.3. The training performance based on EFNN structure (B).

The fourth network predictive model is also based on the proposed algorithm in Figure 3.2 and the data for training are partitioned into 80:10:10. It takes longer time for the training data to reach the convergence point. The training process eventually converged at epochs 140. This is higher than the time it takes the existing network structure to converge. Besides, the best validation performance shown on the training performance graph indicates that, the reduction in training error and timely convergence only improves between 60 – 74% training data.

It should be observed that, as training data increases, some improvement in the training process is recorded and in each instance, validation and testing data reduces. A further reduction from 13% for both validation and testing contributes to poor training performance experienced in the fourth network model represented in Figure 5.4.

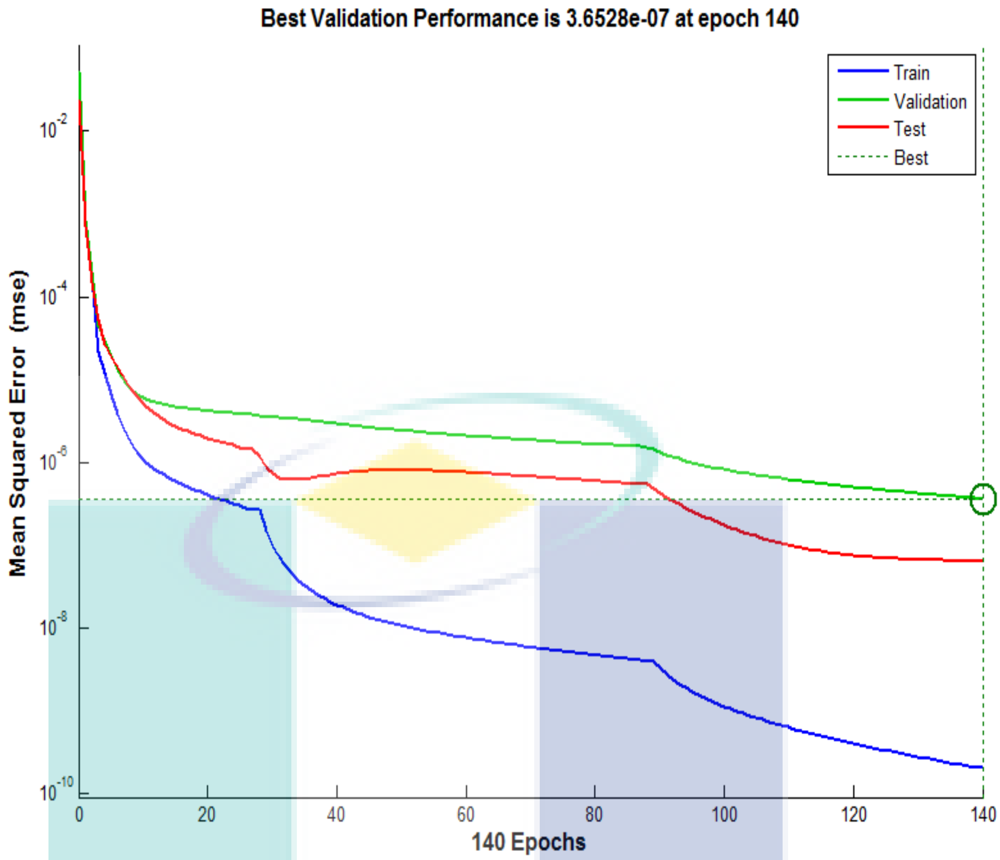

Figure 5.4. The training performance based on EFNN structure (C).

5.3 Results of Evaluating the First Set of Network Predictive Models

All the four network predictive models that are created still need to be evaluated in order to validate the enhancement proposed. This is done by computing the errors associated with each of the network model. Such evaluation involves finding the difference between the predicted outputs and the target outputs, then computing the mean absolute value of these differences (errors). This is sometimes referred to as Mean Absolute Error (MAE).

Evaluation of the network models is very important as this would reveal the error associated with each model that was created. This evaluation requires using a new set of data that has not been trained and in doing this, no target data is required. All the four models that were created using 1200 data based on the existing and the proposed enhanced approach are evaluated. The evaluation involves simulating each of the model with 200 untrained data; thus, the accuracy is measured based on MAE formula represented in Eq. (3.11), in chapter 3.

The four different partitioning of data through which model is fitted and their errors is graphically illustrated in Figure 5.5. For the purpose of making comparisons among the errors computed, it is numerically represented in Table 5.1.

Figure 5.5. Mean Absolute Errors of models from different data partitions.

If an error is low during training, it may not necessarily follow that the error must be low when simulated using untrained data. It is a different thing entirely at the evaluation stage since no target data is provided at this stage. This is noticed when the network models are evaluated and the errors in the previous training performance graphs is compared to the errors computed during the evaluation process for each of the network model. The network model created with 74% training data, 13% validation data and 13% test data appears to have lowest error with best validation performance of 71 epochs as shown in Table 5.1.

Table 5.1
Numerical Representation of Errors from Network Predictive Models Created Based on the First Set of Trained Data

Training set (%)	Validation set (%)	Testing set (%)	Error	Epochs
60	20	20	0.2616	124
64	18	18	0.1762	88
74	13	13	0.0295	71
80	10	10	0.4167	140

In order to establish the consistency of the proposed approach in enhancing the feed-forward neural network on the model it creates, another set of 2500 students' data is transformed and normalized. This data is similar in structure to the data used for training in the previous experiments, but they are not the same. This data is also an excerpt of the records of newly enrolled undergraduate students in the university. With this data, another set of four network predictive models is created using 2000 of this data for training, while the remaining 500 portion of the data is used to evaluate each of the models created to ascertain their accuracy.

The first network model created is based on the existing partition, 60:20:20, the training performance of this model is represented in Figure 5.6. But with the implementation of the algorithm represented in Figure 3.2, three more predictive network models are created. The partition used in creating them are: 66:17:17 ; 74:13:13 and 78:11:11.

Again, the training performance is represented in Figures 5.6 to 5.9 based on the partition used. Before implementing the proposed algorithm, the training show that, the best validation performance is reached at epochs 151 (see Figure 5.6). Although, the training error at this point is not high, but lower errors are recorded when the proposed algorithm is introduced as shown in subsequent graphs.

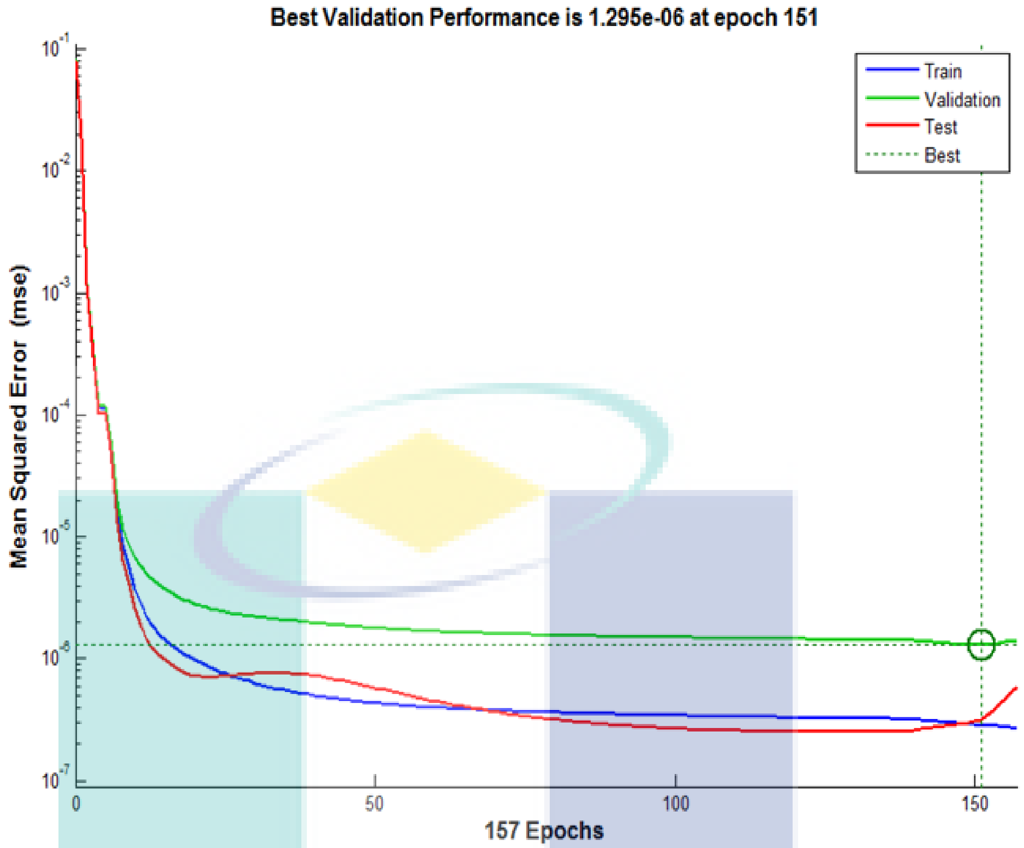

Figure 5.6. The training performance based on FNN structure (2^{nd} set of data).

The second network model is also created based on the data partition 66:17:17; as indicated in the graph; the training, validation and test data look perfectly similar. The best validation performance is reached at epoch 101 and at this point, the error is lower compared to the previous partition in Figure 5.6 that is based on the existing network structure. From the point where the training commences, there is a continuous reduction in error up to the time of convergence. Even at the time when the training process terminates, there was no clear evidence of increase in error. It is also observed that, the training performance shown in Figure 5.7 has the lowest number of epochs among the four network predictive models created using a trained data of 2000.

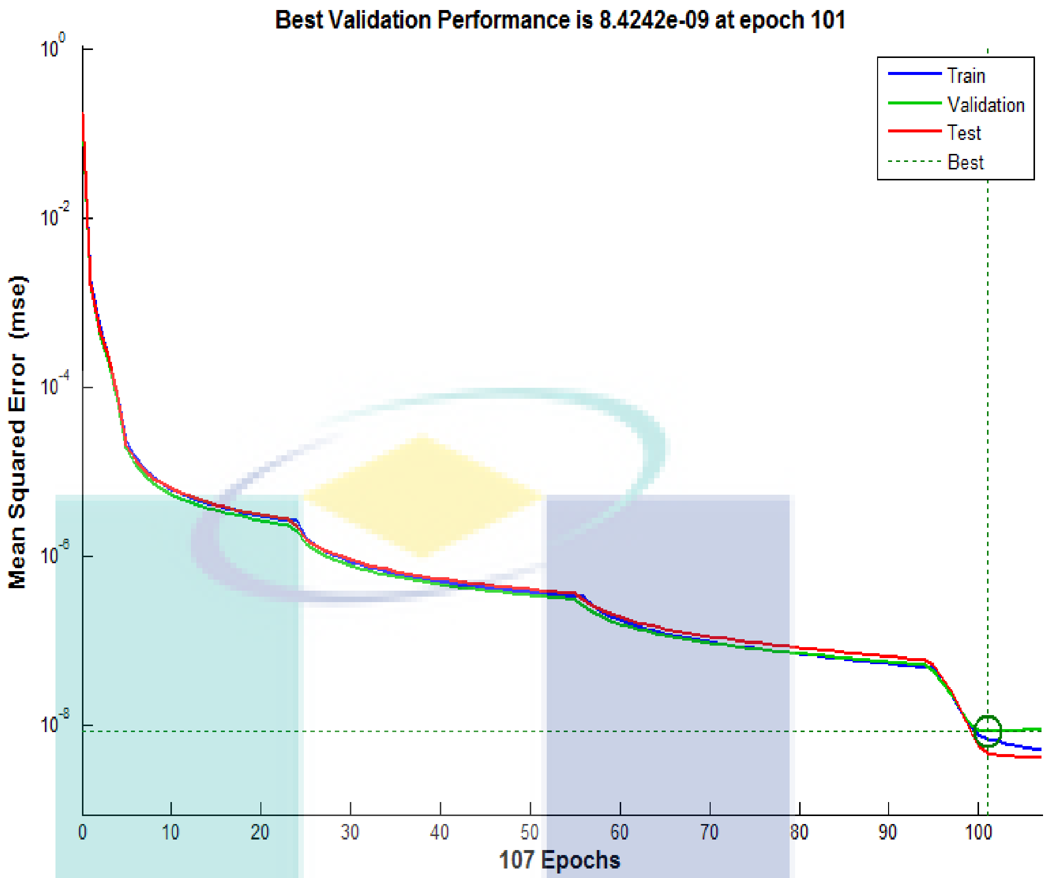

Figure 5.7. The training performance based on EFNN structure (D).

The third network predictive model is created and the data partition conforms to 74:13:13. As the training portion increases from 66% to 74%, the validation and test data decreases accordingly, both must be of equal size. The implication of not making both data to be of equal size is that, test data would be suppressed from the training process and this would definitely affect the resulting outputs. From the training performance shown in Figure 5.8, it takes longer time for the best validation performance to be reached. Although, the error during the training process appears low, the number of epochs is found to be higher than the training performance in Figure 5.7. This implies that, this partition takes longer time to train.

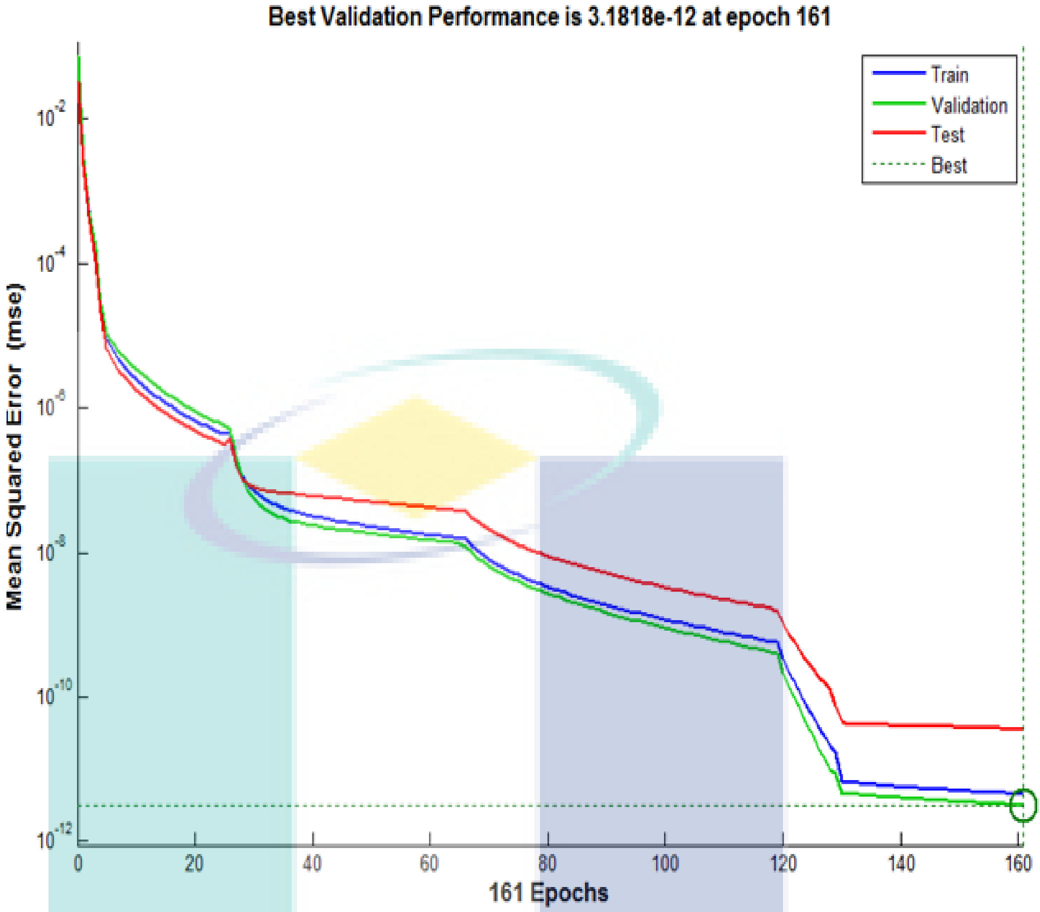

Figure 5.8. The training performance based on EFNN structure (E).

The last and fourth network model created is also based on the proposed enhancement. Like the three previous training performance in Figures 5.6, 5.7 and 5.8, the trained data of size 2000 is partitioned into 78:11:11. As shown in Figure 5.9, it takes so long time to reach the best validation performance of 322 epochs. Although, the training error is not high as shown in the vertical axis of the graph, the training performance cannot be said to be better than the previous partitions. The degree of error associated with a network model can only be properly established when the model is evaluated with a set of untrained data. Sometimes, over-fitting may be responsible for low error during training. This fourth model, has therefore, shown that, the partition used for its creation does not indicate any improvement over the previous partitions.

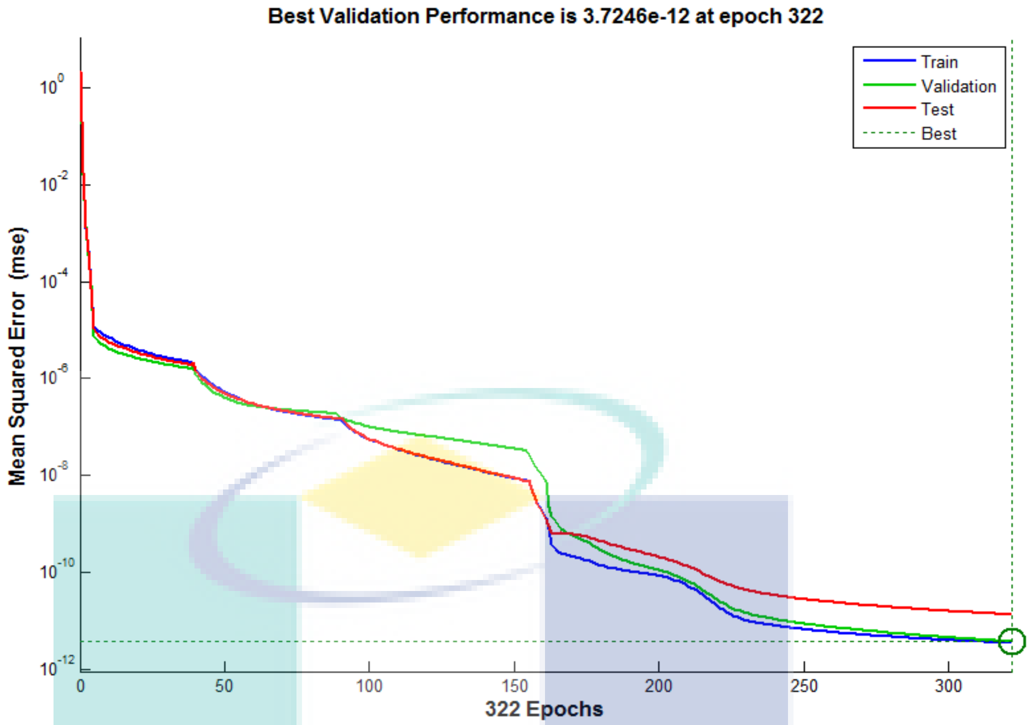

Figure 5.9. The training performance based on EFNN structure (F).

5.4 Results of Evaluating the Second Set of Network Predictive Models

There is need to evaluate the accuracy of all the four network predictive models created as done with the first set of data. This is the only way to establish the consistencies in the accuracy of the algorithm being implemented. When a model is properly trained, the possibility of achieving very low error at the time of training is always very high. However, this may not be the case when the same model is simulated with a set of data the model has never seen. Evaluation is, therefore, an important task to unveil the reliability or otherwise of the model created.

In evaluating these models, similar process used in evaluating the first set of predictive model is also followed here. After the model is trained and subsequently converged; it then produces predicted outputs. The differences between the predicted outputs produced and the target output is determined first, then the Mean Absolute Error (MAE) of these set of values is computed. The results of this computation are shown in Table 5.2.

Table 5.2
Numerical representation of errors from network predictive models created based on the second set of trained data.

Training set (%)	Validation set (%)	Testing set (%)	Error	Epochs
60	20	20	0.00955	151
66	17	17	0.00033	101
74	13	13	0.00085	161
78	11	11	0.11607	322

5.5 Comparison of the Predictive Models Created Using FNN and EFNN

The training performance generated for both FNN and EFNN and the results that emanated from evaluating the errors associated with the models created using both architectures is compared. The comparison revealed that, the division of the dataset for training is one of the effective ways by which the Feed-forward Neural Network technique performance can be enhanced.

The comparison made is based on the error associated with each of the models created when evaluated with a set of untrained data. In the first set of network models created trained with 1200 data, it can be seen that, the second and third models whose training performance is represented in Figures 5.2 and Figure 5.3 respectively have lower error compared to the first model created based on the existing structure. It can also be deduced from Table 5.1 that, the model created using the existing structure has an error of 0.2616, but with the enhanced approach, the error is reduced to 0.0295.

Also, when 2000 data are trained and four more network models are created, evaluation of each model also revealed that, the enhanced EFNN structure brings about a reduction in error. For instance, the first model created with this data which is based on the existing structure has an error of 0.00955, the implementation of the proposed algorithm brings the error down to 0.00033. As the validation and test data reduce further, the model created still has much lower error compared to the error associated with the model created using the existing structure as shown in Table 5.2.

While the existing approach only allows partitioning of the data to 60:20:20, the proposed approach allows more flexibility in the partitioning of the data. In the search for better partitioning of the data in this study, 64:18:18 ; 66:17:17 and 74:13:13 were found to be associated with low errors as shown in Tables 5.1 and 5.2.

5.6 The Resulting Outputs of Implementing the Proposed RBA

In this section, some outputs of the proposed model are represented. Table 5.3 shows the achievements of the newly enrolled students, which is computed based on their historical data. Also shown in the table is the risk associated with their achievements. Two sets of students' data are explored with the proposed RBA, the summarized outputs from each of the results generated is represented and discussed in this section. In order to avoid swelling up of this thesis with list of outputs, only the output generated for all the students using the first set of data is represented in Appendix E.

Table 5.3
Summarized information on students' performance prediction based on the first set of explored data

Performance predicted	Risk Status	Number of Students
Excellent Performance	Risk free	141
Good Performance	Low risk	254
Average Performance	High risk	336
Poor Performance	Very high risk	469

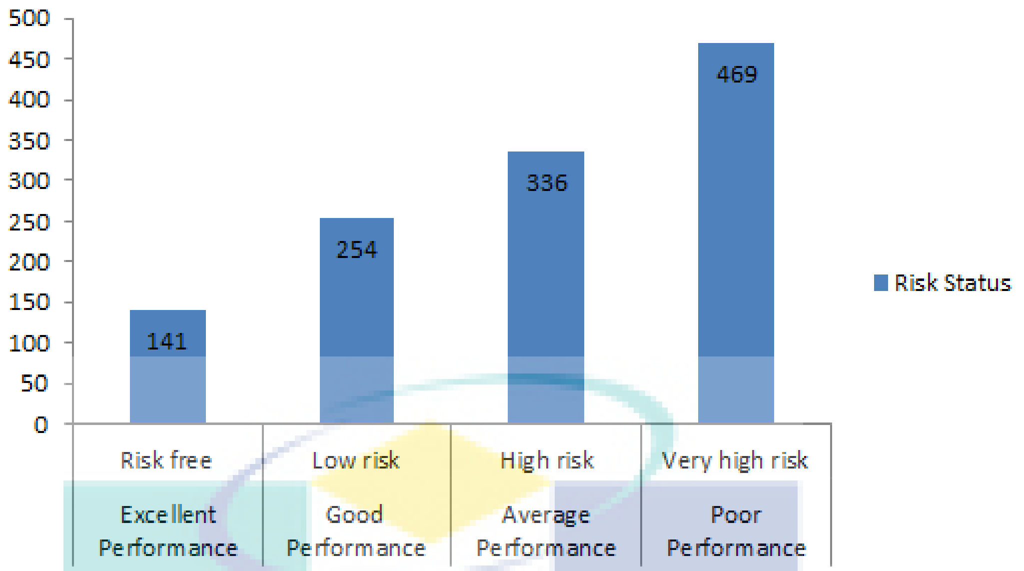

Figure 5.10. Summarized information on the students' performance prediction.

The information shown in Table 5.3 is represented in bar chart for all the 1200 students as shown in Figure 5.10. The students whose performances were predicted as excellent have the lowest number. The trend continues as the status of risk gets higher. Those students whose performances were predicted as poor have the highest number. It can be seen from the table that, students predicted as average, are more in number than those students predicted as good performance.

In order to ascertain the effectiveness of the proposed RBA in the exploration of students' data, the algorithm is further implemented on 2500 data. Putting the output generated in this thesis would swell it up so much, therefore, only the summarized information on the outputs generated is represented in Table 5.4.

As shown in the table, only 251 students are predicted as having excellent performance, this translates to about 10% of the entire data. The poor and average performance students shares equal percentage and they both occupies a large portion of about 35% each, while the performance of 20% of the students are predicted as good. Figure 5.11 shows the detail chart with a summarized information on the number of students, their performance as predicted and the percentage in relation to the data explored.

Table 5.4
Summarized Information on Students' Performance Prediction Based on the Second Set of Explored Data

Performance predicted	Risk Status	Number of Students
Excellent Performance	Risk free	251
Good Performance	Low risk	507
Average Performance	High risk	859
Poor Performance	Very high risk	883

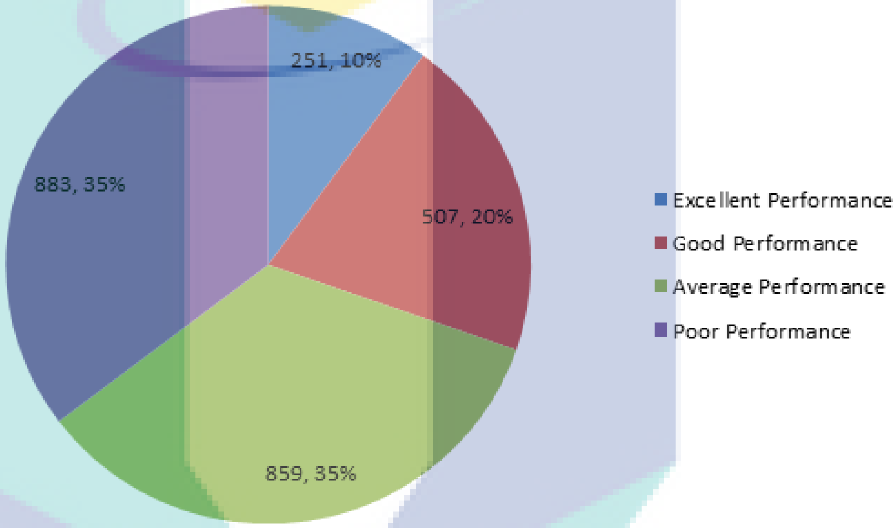

Figure 5.11. Statistical summary of students' performance prediction.

5.7 Results Interpretations

The last stage of data mining process is result interpretation or giving an explanation on the revealed knowledge. The word "knowledge" that is frequently used in data mining has been clarified in the literature, as the structures or patterns that the learning methods produce (Witten et al., 2011).

It is important to add at this point that, predicting the academic performance of students at an early stage of their studentship is not targeted at confirming the number of students that actually failed or passed at the time of graduation as predicted. However, the main motive is for management purposes and to eliminate the possibility of risk.

The data (predictor variables) used for modelling of students' academic performance is already validated if it conforms to what has been reported as predictors in the literature and has the acceptance of domain experts. It is reported in (Suh, 2012) that, predictor variables of high predictive relevance can be suggested by the domain experts.

This research uses the knowledge-driven approach, whereby, the domain expert provides the information that can assist in the interpretation of the outputs generated from the model. In order to effectively use the technique of data mining to arrive at the desired goals, a cooperative effort of humans and computers is very crucial as best results can only be achieved by balancing the knowledge of human experts in describing the problems and goals with the search capabilities of computers (Kantardzic, 2011).

The results of implementing the proposed algorithm that was used to explore the students' historical data is further interpreted for better understanding. Expectedly, the users would want the outputs of the prediction model to be interpretable, comprehensible, and usable. These has been the focus of this research. The summarized information shown in Figure 5.10 distinctly clusters the students' academic performance as: Poor, Avarage, Good and Excellent.

Also, there are risks attached to whatever students has achieved and by extension, their performances. The risk attached to their performance predicted serves as a pointer to their academic stability. The excellent performance students have no risk attached to their performance, this category of students are very few and they are known to be academically sound. These sets of students still need to put in more efforts in order to have their impressive performance sustained.

Very high risk is attached to the students' performance predicted as Poor. Those students predicted as poor or average in performance needs some counselling, encouragement and a proactive intervention. Similarly, category of students that were predicted as Good have low risk attached to their performance, they can always perform much better by mere putting more efforts.

The experimental results of the first approach can be interpreted based the analysis of the errors recorded when the models were simulated with the new set of untrained data. The evaluation carried out clearly differentiates the models in terms of error associated with each of them. If the comparison of the target output and the network predicted output appears almost the same i.e the error is so minimal or negligible, then, it implies that, the accuracy of the model is very high and can be relied upon for prediction. It can therefore, be used to predict subsequent students' academic performance, giving a set of normalized data.

From the results presented in Tables 5.1 and 5.2, the network models created using the partitions 64:18:18; 66:17:17 and 74:13:13 are based on the proposed algorithm in Figure 3.2 and have relatively low errors. These models are therefore suitable and capable of giving accurate predictions.

5.8 Comparisons of the Proposed RBA with some Existing Prediction Techniques

The efficiency of the rule-based algorithm proposed in this research in exploring the students' data for prediction purposes is compared to some related studies reported in the literature. The review of these techniques in chapter 2, revealed what these techniques are capable of doing and what they are not designed to do. Also, they have earlier been implemented on the students' data by several researchers as reviewed in chapter 2. In this research, neural network and the proposed RBA are both implemented. Thus, It is on the basis of the findings that the comparison in Table 5.5 are based.

Specifically, the techniques under considerations are those that are well reported in the literature for the creation of students' academic performance model. One of the methods compared with the proposed algorithm is the technique of fuzzy logic; although, fuzzy logic only based its knowledge representation approach on fuzzy theory, it has also been reported for modelling of students' evaluation (Bai & Chen, 2008), modelling of academic performance evaluation (Yadav & Singh, 2011).

The proposed RBA maintain its accuracy consistently on the data it takes as input, regardless of the size of the data or whether it is seeing the data for the first time or not. However, fuzzy logic lacks the capability to learn, therefore, it cannot generalize whenever it receives set of new input. Generalization is the ability of a model to produce accurate results from the data on which it has not seen (Negnevitsky, 2011).

Although, the proposed RBA is not a learning algorithm, however, it has a centralized and pre-defined set of rules which it uses for mapping input to appropriate outputs that is designated as class labels. Another widely used technique for modelling of academic performance is artificial neural network. It is a typical machine learning method which is widely reported in the literature (Ibrahim & Rusli, 2007a; Oladokun et al., 2008), for modelling of students' academic performance. The proposed RBA based its prediction on clear input-output mapping; this is what transparent process is all about. The transparency of a technique is not measured using formula, but determined through the process that establish the relationship between input and output.

For instance, when the past achievement of a student is computed and found to be 80%, the performance of the student is predicted as excellent. This is because, the rule says student that achieve between 70% and 100 % should be predicted as such. How input become output is thus, followed a clear process, in other word, it is transparent. Using a neural network technique, if we have seven predictors say: [1.42, 1.48, 0.78, 1.00, 1.56, 1.04, 0.96], given that, [0.72] is the target data. After training, [0.71] is produced as the network predicted output. All the processing is done within the hidden layers. It is difficult therefore, to say precisely within the network, how the listed input translates to 0.71 as output predicted. This is why the technique of neural network in mapping the input data to target data is not transparent. Also, this is why the approach is referred to as black box in (Olden & Jackson, 2002) and described as opaque in (Negnevitsky, 2011), which literarily mean not transparent. Most importantly, neural network is an efficient way of creating predictive model.

The proposed RBA is also compared to the use of decision tree in creating classification model of students' academic performance. The research proposed in (Al-Radaideh et al., 2006), applied the decision tree technique to mine students' data. In the process, a system that facilitates the use of the generated rules was built, which allows the final grade of the students to be predicted. The application of decision tree algorithm involves tracing a path from the root to a leaf node, which holds the class prediction for that tuple.

As reported in (Rotshtein & Rakytyanska, 2012), the decision trees can easily be converted to classification rules, that makes its process transparent and easy to understand. To construct a decision tree, with the intent of dividing the individuals of a population into *n* classes, the variable which best separates the individuals of each class must be chosen. According to the precision criterion (García & Mora, 2011), the choice of the variable and the separation condition on this variable depends on the type of tree. Typical decision tree algorithms are: ID3, C4.5 and C5.0.

Similarly, the proposed RBA has some centralized rules on which the students' performance is based. It computes the student's achievement (score), process them and return the performance predicted accordingly. Therefore, both follow similar concept, the input data can easily be traced to the output and vice-versa. However, a decision tree is a machine learning technique and its accuracy is, therefore, tied to how it is properly trained. The enhancement proposed for the use of feed-forward network technique and the use of the proposed RBA have shown much better efficiency in the exploration of students' data for the prediction of their academic performance. The strengths of the proposed approaches as discussed earlier is further summarized in Table 5.4.

Table 5.5

Comparison of the proposed RBA with some Existing Techniques

TECHNIQUES	METRICS		
	Generalization	Accuracy	Transparent process
The Proposed RBA	✔	✔	✔
Neural Networks	✔	✔	x
Decision Tree	✔	✔	✔
Fuzzy Logic	x	✔	✔

Note : ✔ denotes **supported** x denotes **not supported**

The comparison made is summarized in Table 5.5. As shown in the table, the proposed RBA is implemented using sets of past students' data. The outputs the implementation produces is measured and all were found to be correctly classified

(Accurate), using the formular represented in Eq. (4.1). Also, the degree of accuracy recorded when the algorithm is implemented with the first set of data is consistent when another set of similar data is explored (Generalize). Furthermore, tracing all the outputs produced to the input data that produces them, one can understand the rules that connect them as they all conforms to the specified rules and correctly predicted (Transparent process). The first approach proposed to enhance the existing structure of FNN is found to make it generalize much better. But since the whole processing of input data is still confined to the hidden layers, this is why it is widely believed and also described in (Negnevitsky, 2011) that, the process of how input data produces the output using neural network technique is not transparent enough.

CHAPTER 6

CONCLUSIONS

6.1 Introduction

This chapter concludes this thesis. Some of the key findings of this research are discussed. The discussion in this chapter also reiterates some of the contributions of this research to knowledge, especially in the areas of data mining. This chapter also discusses some of the limitations of this research, especially what it is strictly designed to do and what it is not capable of doing. How the proposed model can be put to use is also discussed in this chapter. The brief discussions contain in this chapter has succinctly provided answers to the research problems listed earlier in chapter 1. This chapter also discusses the limitations encountered in the course of this research and the directions for further research are recommended.

6.2 Findings of the Research

This research has demonstrated how the concept of data mining can be used to reveal the useful knowledge embedded in the dataset. The revealed patterns or structure of the data usually helps for decision making. The research has also shown how the neural network techniques, in particular, the feed-forward backpropagation architecture can be enhanced to make it more efficient. Also the proposed rule-based algorithm and its implementation was aimed at devising a means by which past students' data can be optimally explored for decision making. As a result of the enhancement introduced to the existing partitioning technique of the feed-forward network, a significant reduction in the error is noticed when comparison was made with the network model created using the existing structure. The evaluation carried out on the models and the graphical representations of the training performance also reflects that, the models created did not over-fit.

It is also noticed in the course of this research that, the enhancement proposed makes the network to train a bit faster. With the first set of 1200 data that was used to train the network, the best validation performance was reached at epochs 124 with the existing technique (see Figure 5.1); while with the proposed enhancement, the best validation performance was reached at epochs 88 (see Figure 5.2). Also, when the network was trained with 2000 data, similar consistency was recorded, where the best validation performance was reached at epochs 151 and thus, reduce to 101 with the proposed enhancement (see Figures 5.6 and 5.7).

Also, As part of findings during this research, it was noticed that, the feed-forward neural networks have some drawbacks in the modelling of students' data for prediction purposes. One major drawback noticed is that, the operation within the hidden layers does not follow a transparent process; as the operations therein, appears too secrecy. This is capable of making the troubleshooting a bit difficult, especially when the network output deviates widely from the actual outputs already provided as a target. This further necessitates the need to develop the proposed rule-based algorithm for better efficiency. The rules built into the proposed algorithm bring about an easy understanding of how an input transforms to the output. The algorithm classifies, infers and gives a statistical summary of the students' academic performance as predicted.

In order to determine the efficiency of the proposed rule-based algorithm, it was compared to other established techniques that modelled students' academic performance for prediction purposes. In this research, both neural network and the proposed rule-based algorithm were used to explore similar sets of data; their performance can therefore be compared. Other techniques that are well reported in the literature for modelling of students' academic performance apart from neural networks include: decision tree and fuzzy logic. The comparisons of the proposed rule-based algorithm with these techniques for the creation of a predictive model for students' academic performance shows that, the proposed RBA can efficiently be used in lieu of other established methods. The proposed algorithm supports major features expected to produce a reliable predictive model. Such features or metrics include: generalization, accuracy and transparent process. Findings from the literature have shown that not all these features are supported by fuzzy logic technique.

6.3 Research Limitations

In this research, a network predictive model is created and was found to be of high accuracy and most importantly, the models were found to have generalizes well. However, the model created using the historical data of the newly enrolled students cannot be simulated using different forms of data. In data mining, a model can only be evaluated using a set of similar data from which it was created. Testing of the network predictive model created in this research is therefore, limited to the likes of the data through which they were built, such as students' data.

In the second approach, where the exploration of data was based on a certain number of rules; the students' performance that was predicted strictly complies to these rules, therefore, prediction using the proposed algorithm is limited to those inbuilt rules. The algorithm as it is presently listed, cannot predict what it is outside its rules. The rules and by extension, the algorithm is designed to be scalable in such a way that, it can easily be modified to suit users preference.

6.4 Significance Contributions of the Research

This research has contributed to knowledge in the field of data mining. The use of data mining involves fitting the predictive or descriptive models from data with a view to unveiling their useful patterns. This research only focus on creating a predictive model of students' data.

In order to efficiently reveal the patterns embedded in the students' data, this research proposed an enhancement of feed-forward neural network. This is aimed at boosting the performance of network predictive model created using this network structure. The enhancement became necessary in order to alleviate the problem of over-fitting that is peculiar to learning algorithm, such as neural networks. The enhancement introduced reduces the problem of over-fitting to the barest minimum. This was achieved by modifying the function that is responsible for the partitioning of data before the training process commences.

Specifically, the divide function called *dividerand* that normally divides data into the ratio 60:20:20 for training, validation and testing respectively in feed-forward neural networks were modified. The modification of the said function paves way for division of data for training based on the size of data available for exploration. The algorithm that was implemented for the modification is represented in Figure 3.2, while the code for the implementation of the algorithm is listed in Appendix B of this thesis.

The enhancement proposed makes the training dataset to adequately represent the entire data in the input space. In other word, the training data were able to span through the predictor attributes which consequently helps in generalizing. With the proposed enhancement, the partitioning of data for training purpose can be based on the size of the available data to be trained, and not necessarily a fixed percentage as in the existing FNN.

The present research also contributes to knowledge through the rule-based algorithm that is proposed for efficient exploration of students' data. The design of the algorithm was made to be simple to adapt. It is also found to be accurate; it follows a transparent process and most importantly, it generalizes well. Comparison of this algorithm with other algorithms that perform similar tasks revealed that, the proposed rule-based algorithm can be used as an alternative technique for use in the creation of prediction model for students' academic performance.

Also, the approaches of data mining proposed in this research, has been able to unveil some useful knowledge embedded in the students' historical data. This sets of data were captured during the time students were seeking for admission. The exploration of this category of data was necessary due to its neglect immediately the admission process is concluded in the institutions of learning. The present research has shown that, information regarding the students' antecedent, especially, their past academic performance can be predicted through the proper exploration of this set of data.

6.5 Applications of the Model

Model has been described as an abstraction of some aspect of a problem (Blaha, 2010). The prediction model created for students' academic performance in this research was aimed at unveiling useful patterns embedded in the students data for decision making. The accuracy recorded through the proposed techniques for the creation of the models is an indication that the model is reliable and would perform optimally to achieve the desired goals.

A model design for the prediction of students' academic performance at the early stage of their studies can unveil several useful information for a sustainable educational growth and adequate planning. This is because, being able to identify the likely victim of drop-out student or those that are prone to over stay on their programme of studies would help to target intervention programmes directly to those students seriously in need. The implication of such timely intervention can lead to students being more focused on their studies and can trigger an exponential increase of graduate students with good grades.

The developed model also assesses the prior academic achievement of students and uses the information to predict their academic performance. Also the predictive model is capable of offering useful information and numerous opportunities for instructors and decision makers. If the teacher is aware of the strengths and weaknesses of the students, such knowledge would guide on the instructional and teaching strategy to adopt that would be impactful to students.

Generally, model developed using the concept of data mining may either be for the purpose of descriptions or predictions. The model developed in this research focuses on predictions; the first part of the research deals with fitting network predictive model from students' historical data, this is a regression task. The second part classifies the same data with a view to predicting their academic performance based on their past achievements. Apart from using the model developed for prediction purposes, it is also capable of classifying and revealing an accurate summary of the analysed data.

6.6 Recommendations for Future Research

This research has shown how a learning algorithm can be enhanced for better performance in the predictive modelling of students' academic performance. The thesis has also revealed how an algorithm designed and which is based on a number of centralized pre-defined rules can be used for efficient exploration of students' data. The research has also shown how the knowledge embedded in the historical data of the newly enrolled undergraduate students can be unveiled. This research can further be improved upon.

The enhanced FNN has been used to create a predictive model which results in better accuracy and network model that generalizes well. The target attribute is continuous, which is an example of a regression task. Neural networks do not generate rules, but in order to make the model work as a classifier, further work needs to be done by discretizing the target data. Also, the neural networks can perform more useful tasks if combined with other technique such as fuzzy logic for easy classification of data.

The creation of predictive models in this research used the concept of early stopping in neural networks for the training of the data sets. In this approach, the available data were divided into three subsets: training, validation and testing. This is one of the possible ways of improving generalization in neural network. The performance of the prediction model created in this research using the feed-forward neural network can be improved further by exploring the objective function towards giving an optimized results. This function used to be the sum of squares of the network errors on the training set, this is otherwise known as regularization. Such regularization can lead to an optimal solution during the training process.

REFERENCES

Abdous, M. h., He, W., & Yen, C.-J. (2012). Using data mining for predicting relationships between online question theme and final grade. *Journal of Educational Technology & Society, 15*(3), 77-88.

Al-Radaideh, Q. A., Al-Shawakfa, E. M., & Al-Najjar, M. I. (2006). *Mining student data using decision trees.* Paper presented at the International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan.

Ampazis, N., & Perantonis, S. J. (2002). Two highly efficient second-order algorithms for training feedforward networks. *Neural Networks, IEEE Transactions on Neural networks, 13*(5), 1064-1074.

Anderson, P. (2015). Accuracy and Precision. Retrieved 20 February, 2016, from https://www.sophia.org/tutorials/accuracy-and-precision--3

Andrews, R., Diederich, J., & Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems, 8*(6), 373-389.

Arora, N., & Saini, J. (2013). A fuzzy probabilistic neural network for student's academic performance prediction. *International Journal of Innovative Research in Science, Engineering and Technology, 2*(9), 4425-4432.

Arsad, P. M., Buniyamin, N., & Ab Manan, J.-l. (2014). *Neural Network and Linear Regression methods for prediction of students' academic achievement.* Paper presented at the Global Engineering Education Conference (EDUCON), IEEE.

Arsad, P. M., Buniyamin, N., & Manan, J.-l. A. (2012). *Neural network model to predict electrical students' academic performance.* Paper presented at the 4th International Congress on Engineering Education (ICEED).

Arsad, P. M., Buniyamin, N., & Manan, J.-l. A. (2013). *Prediction of engineering students' academic performance using Artificial Neural Network and Linear Regression: A comparison.* Paper presented at the 2013 IEEE 5th Conference on Engineering Education (ICEED).

Bai, S.-M., & Chen, S.-M. (2008). Evaluating students' learning achievement using fuzzy membership functions and fuzzy rules. *Expert Systems with Applications, 34*(1), 399-410.

Baker, R. (2010). Data mining for education. *International encyclopedia of education, 7*, 112-118.

Bakshi, U. A., & Bakshi, A. V. (2009). *Instrumentation* (third ed.). India: Technical Publications Pune.

Baradwaj, B. K., & Pal, S. (2011). Mining educational data to analyze students' performance. *International Journal of Advanced Computer Science and Applications, 2 (6)*.

Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students' performance. *arXiv preprint arXiv:1201.3417*.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2010). Neural Network Toolbox 7. *User's Guide, MathWorks*.

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., . . . Wiswedel, B. (2008). *KNIME: The Konstanz information miner*: Springer.

Black, N. T., & Ertel, W. (2011). *Introduction to artificial intelligence*: Springer Science & Business Media.

Blaha, M. (2010). *Pattern of Data Modeling*. USA: CRC Press Taylor & Francis.

Brown, K. S. (2009). *Factors that Predict Academic Achievement for Students Who are Undecided Majors.* Virginia Polytechnic Institute and State University Blacksburg, Virginia.

Bunkar, K., Singh, U. K., Pandya, B., & Bunkar, R. (2012). *Data mining: Prediction for performance improvement of graduate students using classification.* Paper presented at the Ninth International Conference on Wireless and Optical Communications Networks (WOCN).

Chan, C.-C. (2007). *A Framework for Assessing usage of web-based E-Learning systems.* Paper presented at the 2nd International Conference on Innovative Computing, Information and Control. .

Chen, J.-F., & Do, Q. H. (2014). Training Neural Networks to Predict Student Academic Performance: A comparison of Cuckoo search and Gravitational search Algorithms. *International Journal of Computational Intelligence and Applications, 13*(01).

Chisholm, A. (2013). *Exploring Data with RapidMiner*. UK: Packt Publishing.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine learning, 15*(2), 201-221.

Coskun, N., & Yildirim, T. (2003). *The effects of training algorithms in MLP network on image classification.* Paper presented at the Proceedings of the International Joint Conference on Neural Networks.

Damez, M., Dang, T. H., Marsala, C., & Bouchon-Meunier, B. (2005). *Fuzzy decision tree for user modeling from human-computer interactions.* Paper presented at the Proceedings of the 5th International Conference on Human System Learning, ICHSL.

DeBerard, M. S., Spielmans, G., & Julka, D. (2004). Predictors of academic achievement and retention among college freshmen: A longitudinal study. *College student journal, 38*(1), 66-80.

Do, Q. H., & Chen, J.-F. (2013). A neuro-fuzzy approach in the classification of students' academic performance. *Computational intelligence and neuroscience, 2013 (6)*.

Erkaymaz, O., ÖZER, M., & Yumusak, N. (2014). Impact of small-world topology on the performance of a feed-forward artificial neural network based on 2 different real-life problems. *Turkish Journal of Electrical Engineering & Computer Sciences, 22*(3).

Fakeye, D. (2014). English language proficiency as a predictor of academic achievement among EFL students in Nigeria. *Journal of Education and Practice, 5*(9), 38-41.

Fei, H., Yao, H.-F., & Ling, Q.-H. (2012). An improved extreme learning machine based on particle swarm optimization *Bio-inspired computing and applications* (pp. 699-704): Springer.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research, 3*, 1289-1305.

Fullér, R. (2013). *Introduction to neuro-fuzzy systems* (Vol. *2*): Springer Science & Business Media.

García, E. P. I., & Mora, P. M. (2011). *Model Prediction of Academic Performance for First Year Students.* Paper presented at the 10th Mexican International Conference Artificial Intelligence (MICAI).

Gu, P., & Zhou, Q. (2012). Student Performances Prediction Based on Improved C4. 5 Decision Tree Algorithm *Emerging Computation and Information teChnologies for Education* (pp. 1-8): Springer.

Haimowitz, I. J., Murren, B. T., Lander, H., Pierce, B. A., & Phillips, M. C. (1999). Generating rules for matching new customer records to existing customer records in a large database: Google Patents.

Han, J., Kamber, M., & Pei, J. (2012). *DATA MINING Concepts and Techniques* (3rd ed.): Morgan Kaufman, Elsevier, USA.

Hassoun, M. (2008). *Fundamentals of Artificial Neural Networks*. USA: MIT Press.

Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd Ed.). New Jersey: Pearson Education, Inc.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *The Computing Research Repository (CoRR)*

Ibrahim, Z., & Rusli, D. (2007a). *Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression.* Paper presented at the 21st Annual SAS Malaysia Forum.

Ibrahim, Z., & Rusli, D. (2007b). *Predicting students" Academic Performance: Comparing Artificial Neural Network, Decision Tree and Linear Regression.* Paper presented at the Proceedings of the 21º Annual SAS Malaysia Forum, Kuala Lumpur, Malaysia.

Issanchou, S., & Gauchi, J. P. (2008). Computer-aided optimal designs for improving neural network generalization. *Neural Netw, 21*(7), 945-950.

Jin, C., Jin, S.-W., & Qin, L.-N. (2012). Attribute selection method based on a hybrid BPNN and PSO algorithms. *Applied Soft Computing, 12*(8), 2147-2155.

Jukic, N., Vrbsky, S., & nestorov, S. (2014). *Database Systems: Introduction to Database and Data warehouses*: Pearson.

Kantardzic, M. (2011). *DATA MINING: Concepts, Models, Methods and Algorithms* (2nd ed.). New Jersey: IEEE Press, John Wiley & Sons, Inc.

Klawonn, F., Nauck, D., & Kruse, R. (1995). *Generating rules from data by fuzzy and neuro-fuzzy methods.* Paper presented at the In: Proc. Fuzzy-Neurosystems '95, DARMSTADT.

Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (Vol. *4*): Prentice hall New Jersey.

Kumar, T. S. (2014). *Introduction to Data Mining* (First ed.): Pearson.

Kwork, T., & Yeung, D. (1999). Constructive algorithms for structure learning in feed-forward neural networks for regression problems: A survey. *IEEE Transactions on Neural Networks, 8*(3).

Ledolter, J. (2013). *Data Mining and Business Analytics with R.* Canada: John Wiley & Sons Inc.

Lin, C.-L., Wang, J., Chen, C.-Y., Chen, C.-W., & Yen, C. (2009). Improving the generalization performance of RBF neural networks using a linear regression technique. *Expert Systems with Applications, 36*(10), 12049-12053.

Linoff, G. S., & Berry, M. J. (2011). *Data mining techniques: for marketing, sales, and customer relationship management*: John Wiley & Sons.

Lopez, M. I., Luna, J., Romero, C., & Ventura, S. (2012). Classification via Clustering for Predicting Final Marks Based on Student Participation in Forums. *International Educational Data Mining Society*.

Lye, C.-T., Ng, L.-N., Hassan, M. D., Goh, W.-W., Law, C.-Y., & Ismail, N. (2010). Predicting Pre-university student's Mathematics achievement. *Procedia-Social and Behavioral Sciences, 8*, 299-306.

Maimon, O., & Rokach, L. (2008). *Soft Computing for Knowledge Discovery and Data Mining*: Springer Science and Business Media, LLC.

Maldonado, S., & Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences, 179*(13), 2208-2217.

MathWorks. (2015a). *Fuzzy Logic Toolbox User's Guide*. USA: The MathWorks, Inc.

McKenzie, K., & Schweitzer, R. (2001). Who succeeds at university? Factors predicting academic performance in first year Australian university students. *Higher education research and development, 20*(1), 21-33.

Mishra, T., Kumar, D., & Gupta, S. (2014). *Mining Students' Data for Prediction Performance.* Paper presented at the Fourth International Conference on Advanced Computing & Communication Technologies (ACCT).

Mohammad, Y. H., & Almahmeed, M. A. (1988). An evaluation of traditional admission standards in predicting Kuwait University students' academic performance. *Higher education, 17*(2), 203-217.

Munakata, T. (2008). *Fundamentals of the new Artificial Intelligence* (Second ed.). London: Springer-Verlag

Naik, B., & Ragothaman, S. (2004). Using neural networks to predict MBA student success. *College Student Journal, 38*(1), 143.

Narayan, S., Tagliarini, G. A., & Page, E. W. (1996). Enhancing MLP networks using a distributed data representation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26*(1), 143-149.

Negnevitsky, M. (2011). *Artificial Intelligence A Guide to Intelligent Systems*: Pearson Education Limited.

Nie, P.-y. (2005). A filter method for solving nonlinear complementarity problems. *Applied mathematics and computation, 167*(1), 677-694.

Oladokun, V., Adebanjo, A., & Charles-Owaba, O. (2008). Predicting students' academic performance using artificial neural network: A case study of an engineering course. *The Pacific Journal of Science and Technology, 9*(1), 72-79.

Olden, J. D., & Jackson, D. A. (2002). Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling, 154*(1), 135-150.

Pechenizkiy, M., Puuronen, S., & Tsymbal, A. (2008). Does Relevance Matter to Data Mining Research? *Data Mining: Foundations and Practice* (pp. 251-275): Springer.

Piccolo, S. R., & Frey, L. J. (2012). ML-Flex: A flexible toolbox for performing classification analyses in parallel. *The Journal of Machine Learning Research, 13*(1), 555-559.

Rainardi, V. (2008). *Building a Data Warehouse With Examples in SQL Server*: Apress.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40*(6), 601-618.

Romero, C., Ventura, S., Espejo, P. G., & Hervás, C. (2008). *Data mining algorithms to classify students.* Paper presented at the Educational Data Mining 2008.

Rotshtein, A. P., & Rakytyanska, H. B. (2012). *Fuzzy Evidence in Identification Forecasting and Diagnosis*: Springer.

Ruggieri, S. (2002). Efficient C4. 5 classification algorithm. *IEEE Transactions on Knowledge and Data Engineering, 14*(2), 438-444.

Scheuer, O., & McLaren, B. M. (2012). Educational data mining *Encyclopedia of the Sciences of Learning* (pp. 1075-1079): Springer.

Sedgewick, R., & Flajolet, P. (2013). *An Introduction to the Analysis of Algorithms* (2nd ed.). USA: Addison Wesley.

Sharabiani, A., Karim, F., Atanasov, M., & Darabi, H. (2014). *An enhanced bayesian network model for prediction of students' academic performance in engineering programs.* Paper presented at the Global Engineering Education Conference (EDUCON).

Shinghal, R. (2013). *Introduction to Fuzzy logic*. Delhi: PHI Learning Private Limited.

Simard, D., Nadeau, L., & Kröger, H. (2005). Fastest learning in small-world neural networks. *Physics Letters A, 336*(1), 8-15. doi: 10.1016/j.physleta.2004.12.078

Skiena, S. S. (2008). *The Algorithm Design Manual* (2nd ed.). London: Springer-Verlag.

Stanley, K., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation, 10*(2), 99-127.

Suh, S. C. (2012). *Practical Applications of DATA MINING*: Jones & Bartlett Learning.

Sundar, P. P. (2013). A Comparative Study For Predicting Student's Academic Performance Using Bayesian Network Classifiers. *IOSR Journal of Engineering (IOSRJEN) e-ISSN*, 2250-3021.

Tan, M., & Shao, P. (2015). Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method. *International Journal of Emerging Technologies in Learning (iJET), 10*(1), pp. 11-17.

Taylan, O., & Karagözoğlu, B. (2009). An adaptive neuro-fuzzy model for prediction of student's academic performance. *Computers & Industrial Engineering, 57*(3), 732-741.

Thiele, T., Singleton, A., Pope, D., & Stanistreet, D. (2014). Predicting students' academic performance based on school and socio-demographic characteristics. *Studies in Higher Education*, 1-23.

Thimm, G., & Fiesler, E. (1995). *Evaluating prunning methods.* Paper presented at the International Symposium on Artificial Neural Networks, Taiwan, Republic of China.

Trockel, M. T., Barnes, M. D., & Egget, D. L. (2000). Health-related variables and academic performance among first-year college students: implications for sleep and other behaviors. *Journal of American college health, 49*(3), 125-131.

Tuffery, S. (2011). *Data Mining and Statistics for Decision Making*. USA: John Wiley & Sons Ltd.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(2579-2605), 85.

Vandamme, J. P., Meskens, N., & Superby, J. F. (2007). Predicting academic performance by data mining methods. *Education Economics, 15*(4), 405-419.

Vieira, J., Dias, F. M., & Mota, A. (2004). *Neuro-fuzzy systems: a survey.* Paper presented at the 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italia.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research, 30*(1), 79.

Win, R., & Miller, P. W. (2005). The effects of individual and school factors on university students' academic performance. *Australian Economic Review, 38*(1), 1-18.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *DATA MINING Practical Machine Learning Tools and Techniques* (3rd Edition ed.): Morgan Kaufmann.

Wu, J., & Coggeshall, S. (2012). *Foundations of Predictive Analytics*. USA: CRC Press Taylor & Francis Group.

Yadav, R. S., & Singh, V. P. (2011). Modeling academic performance evaluation using soft computing techniques: a fuzzy logic approach. *International Journal on Computer Science and Engineering, 3*(2), 676-686.

Yam, J. Y., & Chow, T. W. (2001). Feedforward networks training speed enhancement by optimal initialization of the synaptic coefficients. *Neural Networks, IEEE Transactions on, 12*(2), 430-434.

Zhao, G., Shen, Z., Miao, C., & Man, Z. (2009). *On improving the conditioning of extreme learning machine: a linear case.* Paper presented at the Information, Communications and Signal Processing. ICICS 2009.

APPENDIX A

LIST OF PUBLICATIONS

The following publications had been made out of this thesis.

Journals

1. Adeleke Raheem Ajiboye, Ruzaini Abdullah-Arshah, Hongwu Qin (2015), Using An Enhanced Feed-forward BP Network for Predictive Model Building from Students' Data; *Intelligent Automation & Soft-Computing*, Taylor & Francis Group. DOI: http://dx.doi.org/10.1080/10798587.2015.1079364

2. Adeleke Raheem Ajiboye, Ruzaini Abdullah-Arshah, Hongwu Qin (2015), A Novel Approach to Efficient Exploration and Mining of Students' Data; *Journal of Theoretical and Applied Information Technology,* Vol. 79 Issue 1, p176-184.

3. A.R. Ajiboye, R. Abdullah-Arshah, H. Qin and H. Isah-Kebbe (2015), Evaluating The Effect of Dataset Size on Predictive Model Using Supervised Learning Technique; *International Journal of Software Engineering & Computer Systems,* vol. 1, pp. 75-84.

4. Adeleke Raheem Ajiboye, Ruzaini Abdullah-Arshah, Hongwu Qin (2013), Prediction and Modelling of Students' Academic Achievement - A fuzzy logic approach; *Research Inventy: International Journal of Engineering and Science*, Vol.3, Issue 11.

5. Adeleke Raheem Ajiboye, Ruzaini Abdullah Arshah, Hongwu Qin, Jamila Abdulhadi (2016), Comparing the Performance of Predictive Models Constructed Using the Techniques of Feed-forward and Generalized Regression Neural Networks; *International Journal of Software Engineering & Computer Systems,* vol. 2, pp. 66 – 73.

Conference papers presented

1. Adeleke Raheem Ajiboye, Ruzaini Abdullah Arshah and Hongwu Qin (2015), Using an Enhanced Feed-Forward Neural Network Technique for Prediction of Students' Performance; *3rd International Conference on Computer Science and Data Mining* (ICCSDM'15), Dubai, UAE, May 20-21.

2. Raheem Ajiboye, Ruzaini Abdullah Arshah and Hongwu Qin, Jamila Abdulhadi (2015), Comparing the Performance of Predictive Models Constructed Using the Techniques of Feed-forward and Generalized Regression Neural Networks; *National Conference on Post Graduate Studies, (NCON '15)*, Universiti Malaysia Pahang, Kuantan, Jan. 25-26.

APPENDIX B

CODE LISTING FOR DYNAMIC PARTITIONING OF DATASET

```
% Implimentation done with the Matlab Codes
% Upload the predictors to workspace
clc
input1= sne;
input2= hsr;
input3= nhsea;
input4= age;
input5= sti;
input6= yle;
input7= so;
tagt=target;
   tgt= tagt;
inpt=[input1,input2,input3,input4,input5,input6,input7];
   % transpose the input and the target data
inpt=inpt'
tgt=tgt'
   % create the network model and make the partitioning of dataset
flexible
mynewnet = newff(inpt,tagt,10);
mynewnet.divideFcn=' ';
mynewnet.divideFcn='divideind';
trainInd=1:888;
valInd=889:1044;
testInd=1045:1200;
mynewnet.divideParam.trainInd=1:888;
mynewnet.divideParam.valInd=889:1044;
```

```
mynewnet.divideParam.testInd=1045:1200;

[traininpt,valinpt,testinpt]=divideind(inpt,trainInd,valInd,testInd);

    % parameters and properties setting

mynewnet.trainParam.epochs= 800;

mynewnet.trainParam.goal= 0;

mynewnet.trainParam.min_grad=1e-7;

mynewnet.trainParam.max_fail= 10;

mynewnet.trainParam.mu=0.0001;

mynewnet.trainParam.mu_dec=0.1;

mynewnet.trainParam.mu_inc=10;

    % train the network

mynewnet=train(mynewnet,inpt,tagt)

    % plot the training performance

mynewnet.trainPerformance.show

    % import new set of untrained data to simulate the network model

simudat=[sne,hsr,nhsea,age,sti,yle,so];

simudat=simudat';

mynewnet_simu_outputs=mynewnet(simudat)

mynewnet_simu_outputs'

        %  plot the simulated outputs

stem(TAGT);figure(gcf);

stem(mynewnet_simu_outputs);figure(gcf);

   % using nntool to create the FNN network model named mynewnet2

  % and simulate it with the same data
mynewnet2_simu_outputs=mynewnet2(simudat)
mynewnet2_simu_outputs'
stem(mynewnet2_simu_outputs);figure(gcf);
```

APPENDIX C

IMPLEMENTATION OF THE PROPOSED RULE-BASED ALGORITHM

```
//create a database and a table structure

// connect to the database

<?php

$host = "localhost"; $user = "root"; $password = ""; $dbname =
"student_performance";

$con = @mysql_connect($host, $user, $password) or die("can’t connect
to the database, sorry for any inconvenience!");

@mysql_select_db($dbname) or die("Database not found!");

?>

// upload data to be explored

<body>

</p>

<form enctype="multipart/form-data" name="frmdata" id="frmdata"
action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">

<table width="80%" border="1" cellpadding="0" cellspacing="0">

  <tr>

    <td width="27%">Browse File</td>

    <td width="73%"><input type="file" name="file" id="file" /></td>

  </tr>

    <td>Maximum Obtainable Score</td>

    <td><input type="text" name="score" id="score" class="txt"/></td>
```

```
 </tr>

 <tr>

   <td colspan="2"><div align="center"><br />

     <input type="submit" name="cmdupload" id="cmdupload" 
value="Upload Data" class="btn"/>

   </div></td>

   </tr>

</table>

</form>

<br/>

<?php

 if($_REQUEST['cmdupload']!='')

                        {

                  $fname = @$_FILES['file']['name'];

       $ext = @explode(".",$fname);  $ext = $ext[1]; $found=false;

               $max_score=$_REQUEST['score'];

                   if ($ext != "csv" and $ext != "CSV") { ?> <script 
language="javascript">window.alert("Sorry, invalid file format. The file 
to be uploaded must be '.csv' file only"); window.close();</script> <?php 
$found=true; }

     if($found!=true)

               {

$uploadDir = "upload_files/";
```

```
$myFileDate = @date("Ymd") . "_" .
@str_replace(":","",@date("H:i:s"));

$pubfile = "$myFileDate" . "_".$fname;

$pubfile = @str_replace("/","",$pubfile);

$uploadFile = $uploadDir . $pubfile;

 if(@move_uploaded_file($_FILES['file']['tmp_name'], $uploadFile))

 {          // file uploaded

  //$input_file = $uploadFile;

 $total_rec=0;

  $file_array = @file("$uploadFile");

$total_rec=@sizeof($file_array);

@mysql_query("delete from datatb");

@mysql_query("delete from max_scoretb");

@mysql_query("insert into max_scoretb set max_score='$max_score'");

 while (list($line_num, $line) = each($file_array))

{

    // provisions were made for 20 predictors in the table structure

$myLine = @explode(",",$line);        $regno = @trim(@$myLine[0]);

$f1 = @trim(@$myLine[1]);             $f2 = @trim(@$myLine[2]);

$f3 = @trim(@$myLine[3]);             $f4 = @trim(@$myLine[4]);

 $f5 = @trim(@$myLine[5]);             $f6 = @trim(@$myLine[6]);

$f7 = @trim(@$myLine[7]);             $f8 = @trim(@$myLine[8]);

$f9 = @trim(@$myLine[9]);             $f10 = @trim(@$myLine[10]);
```

```
$f11 = @trim(@$myLine[11]);        $f12 = @trim(@$myLine[12]);
 $f13 = @trim(@$myLine[13]);       $f14 = @trim(@$myLine[14]);
 $f15 = @trim(@$myLine[15]);       $f16 = @trim(@$myLine[16]);
$f17 = @trim(@$myLine[17]);        $f18 = @trim(@$myLine[18]);
$f19 = @trim(@$myLine[19]);        $f20 = @trim(@$myLine[20]);
@mysql_query("insert into datatb set
regno='$regno',field1='$f1',field2='$f2',field3='$f3',field4='$f4',field5='$
f5',field6='$f6',field7='$f7',field8='$f8',field9='$f9',field10='$f10',field11
='$f11',field12='$f12',field13='$f13',field14='$f14',field15='$f15',field16
='$f16',field17='$f17',field18='$f18',field19='$f19',field20='$f20'");
        } //end while
echo "<font color='red'><b>The data has been loaded
successfully</b></font>";
        }
        else
echo "<font color='red'><b>Error uploading file</b></font>";
        }        //end of if not found
        }        //end of action posting
?>
</div>
</body></html>
<?php @session_start();?>
<html xmlns="http://www.w3.org/xhtml"><head><title>Student
Performance Prediction</title>
```

```
<link rel="shortcut icon" href="images/logo.jpg"> <!-- put the
image/logo on the browser tab -->

<link rel="stylesheet" type="text/css" href="style.css" />

<script>

function swapcontent(cv,v,a,b,c,d,e,f,g,h,i,j,k,l)

{   //swap content begins where cv means div id name

        var divid="#"+cv;

        $(divid).html('<img src="images/loader.gif" width="100"
height="100" alt="loading">').show();

        $("#roll").html('<img src="images/loader.gif" width="100"
height="100" alt="loading">').show();

        var url="ajax.php";

        var str;

 if(cv=='risk_section') //start view

  {

          var r;

          if(v=='range_high')

      {

        r=prompt("Enter the total record limit");

}
$.post(url,$("form").serialize()+"&contentvar="+cv+"&action="+v+"&r
ange="+r,function(data)

{  //ajaxfile/scriptfile_a is called undernith

$("#display").html(data).show(); //report result from the ajaxfile, the data
stores the information to be displayed from ajaxfile
```

```
$("#roll").html('').show();

$(divid).html('').show();

});

 }

 if(cv=='another')

  {

$.post(url,$("form").serialize()+"&contentvar="+cv,function(data)

{

$(divid).html(data).show();

});

 }          //end if

}         //end of swapcontent

 </script>

          </head>

<body>

<?php

// ############  Header here ############-->

@include("header.php");

// ############  Side Bar here ############-->

 @include("sidebar.php");

 $res_o=@mysql_query("select max_score from max_scoretb");
$rs_o=@mysql_fetch_array($res_o); $max_score=$rs_o['max_score'];

 $res_d=@mysql_query("select * from datatb order by regno");
```

```
$sn=0;

$tb="<table border=1 cellpadding='0' cellspacing='0'
align='center'><tr><td><b>S/N</b></td><td><b>REGISTRATION
NO</b></td><td><b>SCORE ACHEVED</b></td><td><b>RISK
STATUS</b></td></tr>";

            while($rs_d=@mysql_fetch_array($res_d))

{

  ++$sn;

  $regno=$rs_d['regno'];

$total_score=$rs_d['field1'] + $rs_d['field2'] + $rs_d['field3'] +
$rs_d['field4'] + $rs_d['field5'] + $rs_d['field6'] + $rs_d['field7'] +
$rs_d['field8'] + $rs_d['field9'] + $rs_d['field10'] + $rs_d['field11'] +
$rs_d['field12'] + $rs_d['field13'] + $rs_d['field14'] + $rs_d['field15'] +
$rs_d['field16'] + $rs_d['field17'] + $rs_d['field18'] + $rs_d['field19'] +
$rs_d['field20'];

$score_acheive=number_format($total_score/$max_score * 100,2);

$risk=@compute_risk($score_acheive);

  $performance=@compute_performance($score_acheive);

          $tb.="<tr><td>$sn</td><td><b>$regno</b></td><td><b>$score
          _acheive</b></td><td><b>$risk</b></td></tr>";

                      @mysql_query("update datatb set
total_score='$total_score',score_achieved='$score_acheive',risk_status='
$risk',performance_predicted='$performance' where regno='$regno'");

            }      //end of while

            $tb.="</table>";

echo $tb;
```

```
?>

<div id="content">

<form>

<h2><center>Computation of Acheivement and Risk Status</center><!-
-<span class="pubdate"></span>--></h2>

<p>

<div align="center"><input type="button" name="cmdall" id="cmdall"
value="View All" class="btn"
onclick="swapcontent('risk_section','view');"/>   <!--
<input type="button" name="cmdall" id="cmdall" value="View Record
Limit" class="btn" onclick="swapcontent('risk_section','range');"/>--
>   <br/><br/><input type="button" name="cmdall"
id="cmdall" value="View Best Students" class="btn"
onclick="swapcontent('risk_section','range_high');"/>  &nb
sp;<input type="button" name="cmdall" id="cmdall" value="Statistical
Summary" class="btn"
onclick="swapcontent('risk_section','statistics');"/></div><br />

<div id="display"></div>

<div id="roll"></div>

</p>

</form>

</div>
<?php
// function to compute the risk status and predict students' performance
 function compute_risk($score)

{

if($score>=70 and $score<=100)
```

```
            $str="Risk free";
elseif($score>=60 and $score<=69.99)
            $str="Low risk";
elseif($score>=50 and $score<=59.99)
            $str="High risk";
elseif($score<50)
$str="Very high risk";
return ($str);
}
function compute_performance($score)
{
if($score>=70 and $score<=100)
             $str="Outstanding Performance";
elseif($score>=60 and $score<=69.99)
            $str="Good Performance";
elseif($score>=50 and $score<=59.99)
             $str="Average Performance";
elseif($score<50)
            $str="Poor Performance";
 return ($str);
}
function get_total_risk($risk)
{
```

```
$res_c=@mysql_query("select count(*) as total from datatb where
risk_status='$risk'");
  $rs_c=@mysql_fetch_array($res_c);
  return $rs_c['total'];
}
function get_total_std()
{
  $res_c=@mysql_query("select count(*) as total from datatb");
  $rs_c=@mysql_fetch_array($res_c);
  return $rs_c['total'];
}
function get_total_risk_pred($risk)
{
  $res_c=@mysql_query("select count(*) as total from datatb where
performance_predicted='$risk'");
  $rs_c=@mysql_fetch_array($res_c);
  return $rs_c['total'];
}
function get_total_std_pred()
{
  $res_c=@mysql_query("select count(*) as total from datatb");
  $rs_c=@mysql_fetch_array($res_c);
  return $rs_c['total'];
}
```

```
>?

// ajax starts here,  settings were made
@session_start();
 @ini_set('max_execution_time', 60000000000);
 @ini_set("memory_limit", "51200M");
 @require_once('connect.php');
 @require_once('function.php');
 $id=@$_REQUEST['contentvar'];
 $contentvar=$_REQUEST['contentvar'];
 if($id=='risk_section')
 {
$action=@$_REQUEST['action'];
 $r=@$_REQUEST['range'];  // for range of records to display
 if($action=='view')
 {
  $sql="select * from datatb order by regno";
 } //end of view
 if($action=='range_high')
  {
  $sql="select * from datatb where score_achieved >=60 order by
score_achieved desc  limit 0,$r";
 }          //end of view
  if($action=='range_high' or $action=='view')
```

```
{
$res_d=mysql_query($sql) or die(mysql_error());
$sn=0;
        $tb="<table border=1 cellpadding='0' cellspacing='0'
align='center'><tr><td><b>S/N</b></td><td><b>REGISTRATION
NO</b></td><td><b>SCORE ACHIEVED</b></td><td><b>RISK
STATUS</b></td></tr>";
while($rs_d=mysql_fetch_array($res_d))
 {
    ++$sn;
    $regno=$rs_d['regno'];
//$total_score=$rs_d['field1'] + $rs_d['field2'] + $rs_d['field3'] +
$rs_d['field4'] + $rs_d['field5'] + $rs_d['field6'] + $rs_d['field7'] +
$rs_d['field8'] + $rs_d['field9'] + $rs_d['field10'] + $rs_d['field11'] +
$rs_d['field12'] + $rs_d['field13'] + $rs_d['field14'] + $rs_d['field15'] +
$rs_d['field16'] + $rs_d['field17'] + $rs_d['field18'] + $rs_d['field19'] +
$rs_d['field20'];
    $total_score=$rs_d['total_score'];
    $score_acheive=$rs_d['score_achieved'];
    $risk=$rs_d['risk_status'];
    $performance=$rs_d['performance_predicted'];
$tb.="<tr><td>$sn</td><td><b>$regno</b></td><td><b>$score_acheiv
e</b></td><td><b>$risk</b></td></tr>";
    //@mysql_query("update datatb set
total_score='$total_score',score_achieved='$score_acheive',risk_status='
$risk',performance_predicted='$performance' where regno='$regno'");
```

```
  } //end of while

  $tb.="</table>";

  echo $tb;

 }

<div id="content">

<form>

<h2><center>Predict Student Performance</center><!--<span
class="pubdate">3/8/2008</span>--></h2>

<p>

<div align="center"><input type="button" name="cmdall" id="cmdall"
value="View All" class="btn"
onclick="swapcontent('predict_section','view');"/>   
<!--<input type="button" name="cmdall" id="cmdall" value="View
Record Limit" class="btn"
onclick="swapcontent('predict_section','range');"/>--
>   <br/><br/><input type="button" name="cmdall"
id="cmdall" value="View Best Students" class="btn"
onclick="swapcontent('predict_section','range_high');"/>  &
nbsp;<input type="button" name="cmdall" id="cmdall"
value="Statistical Summary" class="btn"
onclick="swapcontent('predict_section','statistics');"/></div><br />

<div id="display"></div>

<div id="roll"></div>

</p>

</form>

</div>

<?php
```

```
?>

</div>

</body></html>

 if($action=='statistics')

  {

 $res_s=@mysql_query("select distinct risk_status from datatb order by risk_status");

$tb="<table border=1 cellpadding='0' cellspacing='0' align='center'><tr><th>RISK   STATUS</th><th>% OF RISK</th><th>NUMBER OF STUDENTS</th></tr>";

$total_student=0;

 while($rs_s=@mysql_fetch_array($res_s))

  {

  $risk=$rs_s['risk_status'];

  $no_of_std=get_total_risk($risk);

  $total_rec=get_total_std();

  $no_of_std_perc=@number_format(($no_of_std/$total_rec) * 100,2);

tb.="<tr><td>$risk</td><td><center>$no_of_std_perc</caption></td><td><center>$no_of_std</center></td></tr>";

  $total_student+=$no_of_std;

}

$tb.="<tr><td colspan='2'><b>  TOTAL STUDENTS</b></td><td><center><b>$total_student</b></center></td></tr>";

$tb.="</table>";
```

```
echo $tb;

}

}     //end of risk computation section

if($id=='predict_section')

{

$action=@$_REQUEST['action'];

$r=@$_REQUEST['range'];

if($action=='view')

{

 $sql="select * from datatb order by regno";

} //end of view

 if($action=='range_high')

 {

$sql="select * from datatb where score_achieved >=60 order by
score_achieved desc limit 0,$r";

 } //end of view

 if($action=='range' or $action=='range_high' or $action=='view')

 {

$res_d=mysql_query($sql) or die(mysql_error());

 $sn=0;

$tb="<table border=1 cellpadding='0' cellspacing='0'
align='center'><tr><td><b>S/N</b></td><td><b>REGISTRATION
NO</b></td><td><b>SCORE ACHIEVED</b></td><td><b>RISK
```

```
STATUS</b></td><td><b>PERFORMANCE
PREDICTED</b></td></tr>";

  while($rs_d=mysql_fetch_array($res_d))

   {

              ++$sn;

              $regno=$rs_d['regno'];

               $total_score=$rs_d['total_score'];

 $score_acheive=$rs_d['score_achieved'];

 $risk=$rs_d['risk_status'];

 $performance=$rs_d['performance_predicted'];

$tb.="<tr><td>$sn</td><td><b>$regno</b></td><td><b>$score_acheiv
e</b></td><td><b>$risk</b></td><td><b>$performance</b></td></tr>
";

              //@mysql_query("update datatb set
total_score='$total_score',score_achieved='$score_acheive',risk_status='
$risk',performance_predicted='$performance' where regno='$regno'");

   }     //end of while



   $tb.="</table>";

   echo $tb;
 } //end of view section
 if($action=='statistics')
 {
 $res_s=@mysql_query("select distinct performance_predicted from
datatb order by performance_predicted");
$tb="<table border=1 cellpadding='0' cellspacing='0'
align='center'><tr><th>PERFORMANCE PREDICTED</th><th>% OF
RISK</th><th>NUMBER OF STUDENTS</th></tr>";
```

```
$total_student=0;
            while($rs_s=@mysql_fetch_array($res_s))
{
  $risk=$rs_s['performance_predicted'];
  $no_of_std=get_total_risk_pred($risk);
  $total_rec=get_total_std_pred();
  $no_of_std_perc=@number_format(($no_of_std/$total_rec) * 100,2);
$tb.="<tr><td>$risk</td><td><center>$no_of_std_perc</caption></td>
<td><center>$no_of_std</center></td></tr>";
  $total_student+=$no_of_std;
}
$tb.="<tr><td colspan='2'><b>  TOTAL
STUDENTS</b></td><td><center><b>$total_student</b></center></td
></tr>";
$tb.="</table>";
echo $tb;
}
 }       //end of predict section
?>
```

APPENDIX D

FIRST DATASET

REG NO.	SNE	HSR	NHSEA	SA	IST	YLE	SO
12CC001	1	14	1	3	2	3	2
12CC002	2	13	2	3	3	2	1
12CC003	4	13	1	3	4	3	1
12CC004	4	23	2	3	2	2	1
12CC005	4	15	2	3	4	3	2
12CC006	2	25	2	3	4	3	1
12CC007	1	14	1	3	1	3	1
12CC008	2	5	2	3	1	2	1
12CC009	4	12	1	3	4	2	2
12CC010	1	18	1	3	4	3	1
12CC011	3	13	1	3	1	1	2
12CC012	4	10	2	3	4	3	1
12CC013	4	19	1	3	4	3	1
12CC014	2	17	1	3	4	3	1
12CC015	4	17	1	3	4	2	1
12CC016	4	8	1	3	1	2	1
12CC017	4	21	1	3	4	3	1
12CC018	4	8	1	3	1	3	1
12CC019	3	13	2	3	2	3	1
12CC020	4	20	2	3	4	3	1
12CC021	1	7	1	3	2	3	1
12CC022	1	13	1	3	3	2	2
12CC023	4	14	1	3	1	2	1
12CC024	1	11	2	3	1	1	1
12CC025	1	7	1	3	1	3	1
12CC026	3	8	1	3	4	3	1
12CC027	4	27	1	3	4	3	1
12CC028	4	24	1	3	2	3	1
12CC029	3	12	1	3	1	3	1
12CC030	4	13	1	3	4	2	1
12CC031	1	12	1	3	4	2	1
12CC032	4	18	1	3	3	3	1
12CC033	3	11	1	3	2	3	1
12CC034	2	8	2	3	1	3	1
12CC035	1	9	1	3	1	3	1
12CC036	3	10	1	3	4	2	1
12CC037	4	23	1	3	4	2	1
12CC038	3	10	1	3	4	3	1
12CC039	1	16	1	3	2	3	1

12CC040	4	14	1	3	4	3	1
12CC041	4	22	1	3	2	2	1
12CC042	4	10	1	3	1	3	1
12CC043	3	15	2	3	4	2	1
12CC044	3	17	1	3	4	2	1
12CC045	4	8	1	3	2	3	1
12CC046	4	21	1	3	4	3	1
12CC047	4	14	1	3	2	3	1
12CC048	4	10	2	3	4	3	1
12CC049	4	25	1	3	4	2	1
12CC050	1	16	1	3	4	3	1
12CC051	4	22	1	3	4	3	2
12CC052	4	7	2	3	1	2	1
12CC053	1	12	1	3	2	3	1
12CC054	4	19	2	3	4	3	1
12CC055	1	12	2	3	2	3	1
12CC056	4	17	1	3	3	3	1
12CC057	3	22	1	3	4	3	1
12CC058	4	18	1	1	4	3	1
12CC059	2	7	1	3	1	3	1
12CC060	1	7	1	3	2	3	1
12CC061	1	11	2	3	1	3	1
12CC062	1	13	1	3	4	3	1
12CC063	1	15	1	3	1	3	1
12CC064	4	16	1	3	4	3	1
12CC065	1	13	1	3	1	3	1
12CC066	1	10	1	3	1	3	1
12CC067	4	27	1	3	4	3	1
12CC068	4	22	1	3	4	3	1
12CC069	4	7	1	3	2	3	1
12CC070	4	10	1	3	2	3	1
12CC071	4	12	1	3	4	2	1
12CC072	4	12	1	3	2	3	1
12CC073	1	8	1	3	3	3	1
12CC074	4	19	1	3	4	3	1
12CC075	4	16	1	3	3	3	1
12CC076	4	26	1	3	4	3	1
12CC077	3	15	1	3	3	2	1
12CC078	1	10	1	3	2	3	2
12CC079	4	9	2	3	1	3	1
12CC080	2	11	1	3	1	3	2
12CC081	1	15	1	3	2	3	1
12CC082	2	11	2	3	2	3	1
12CC083	4	12	1	3	2	3	1
12CC084	4	26	1	3	4	3	1
12CC085	1	10	1	3	2	3	1

12CC086	1	7	1	3	1	2	1
12CC087	3	13	1	3	4	3	1
12CC088	4	10	1	3	4	3	1
12CC089	3	10	1	3	4	3	1
12CC090	2	9	1	3	4	3	1
12CC091	4	13	2	3	2	3	1
12CC092	1	18	1	3	2	3	1
12CC093	2	6	2	3	1	3	1
12CC094	4	15	1	3	4	2	1
12CC095	4	17	1	3	4	3	1
12CC096	1	14	1	3	3	3	1
12CC097	4	12	1	3	1	3	1
12CC098	1	10	1	3	4	3	1
12CC099	4	18	1	3	4	3	1
12CC100	4	19	2	3	1	3	1
12CC101	4	11	1	3	1	3	1
12CC102	2	13	2	3	1	3	1
12CC103	4	14	1	3	2	3	1
12CC104	1	9	1	3	2	3	1
12CC105	4	7	2	3	1	3	1
12CC106	4	10	2	3	1	3	1
12CC107	1	11	1	3	2	3	1
12CC108	3	10	2	3	1	3	1
12CC109	1	13	1	3	1	3	1
12CC110	3	14	2	3	1	3	1
12CC111	2	17	2	3	1	3	1
12CC112	2	14	1	3	2	3	1
12CC113	4	13	2	3	4	3	1
12CC114	1	18	1	3	2	3	1
12CC115	4	20	1	3	4	3	1
12CC116	3	11	1	3	1	3	1
12CC117	4	15	2	3	2	3	1
12CC118	4	18	2	3	4	3	1
12CC119	1	9	2	3	2	3	1
12CC120	4	13	1	3	4	3	1
12CC121	3	20	1	3	1	3	2
12CC122	2	22	1	3	4	3	2
12CC123	4	13	1	3	1	3	1
12CC124	1	10	1	3	2	3	1
12CC125	4	15	1	3	4	3	1
12CC126	1	14	1	3	1	3	1
12CC127	4	13	1	3	4	3	1
12CC128	3	17	1	3	4	3	1
12CC129	1	9	1	3	1	3	1
12CC130	1	16	2	3	2	3	1
12CC131	4	11	1	3	4	3	1

12CC132	3	11	2	3	2	3	1
12CC133	2	12	2	3	4	3	1
12CC134	4	18	1	3	4	3	1
12CC135	1	7	1	3	3	3	1
12CC136	4	6	1	3	2	3	1
12CC137	4	12	1	3	2	3	1
12CC138	4	14	1	3	2	3	1
12CC139	3	8	2	3	1	3	1
12CC140	1	13	1	3	4	3	1
12CC141	4	17	1	3	4	3	1
12CC142	4	22	1	3	4	3	1
12CC143	4	20	1	3	4	3	1
12CC144	1	9	2	3	2	3	2
12CC145	2	19	1	3	3	3	1
12CC146	4	24	1	3	4	3	1
12CC147	4	22	1	3	4	3	1
12CC148	4	25	1	3	4	3	1
12CC149	4	11	1	3	4	3	1
12CC150	3	19	1	3	4	3	1
12CC151	4	19	1	3	4	3	1
12CC152	3	18	1	3	4	3	1
12CC153	4	12	2	3	1	3	1
12CC154	2	15	1	3	4	3	1
12CC155	4	20	1	3	4	3	1
12CC156	1	8	1	3	4	3	1
12CC157	1	7	1	3	1	3	1
12CC158	4	12	1	3	4	3	1
12CC159	4	16	1	3	4	3	1
12CC160	4	14	1	3	4	3	1
12CC161	3	13	1	3	2	3	1
12CC162	4	11	2	3	2	3	1
12CC163	2	12	1	3	3	3	1
12CC164	4	15	1	3	4	3	1
12CC165	4	8	1	3	4	3	1
12CC166	3	16	1	3	4	3	1
12CC167	4	8	2	3	4	3	1
12CC168	4	13	2	3	4	3	1
12CC169	4	8	2	3	1	2	1
12CC170	1	9	2	3	4	2	1
12CC171	1	8	1	3	2	3	1
12CC172	3	8	2	3	3	3	1
12CC173	2	17	1	3	2	3	2
12CC174	4	12	1	3	4	3	2
12CC175	4	15	1	3	3	3	1
12CC176	4	22	1	3	4	3	1
12CC177	3	8	2	3	1	3	1

12CC178	4	10	1	3	3	3	1
12CC179	3	12	1	3	4	3	1
12CC180	4	22	1	3	2	3	1
12CC181	3	9	2	3	2	3	1
12CC182	1	12	2	3	4	3	1
12CC183	2	10	1	3	2	3	1
12CC184	3	14	1	3	4	3	1
12CC185	1	12	1	3	1	3	1
12CC186	4	15	1	3	4	3	2
12CC187	3	15	2	3	4	3	1
12CC188	4	20	1	3	2	3	1
12CC189	4	15	1	3	1	3	1
12CC190	4	20	1	3	4	3	1
12CC191	4	18	1	3	4	3	1
12CC192	4	15	2	3	4	3	1
12CC193	4	20	1	3	3	3	1
12CC194	4	20	1	3	2	3	1
12CC195	4	11	1	3	4	3	1
12CC196	4	12	2	3	4	3	1
12CC197	4	21	1	3	3	3	1
12CC198	4	18	2	3	4	3	1
12CC199	1	8	2	3	1	3	1
12CC200	2	9	1	3	2	3	1
12CC201	4	21	1	3	4	3	1
12CC202	4	10	1	3	4	3	2
12CC203	4	11	1	3	2	3	1
12CC204	4	11	2	3	4	3	1
12CC205	4	16	1	3	3	3	1
12CC206	4	17	1	3	3	3	1
12CC207	2	10	1	3	2	3	1
12CC208	1	9	1	3	1	3	1
12CC209	4	10	1	3	1	3	1
12CC210	4	13	1	3	4	3	1
12CC211	4	14	1	3	3	3	1
12CC212	4	14	1	3	2	3	1
12CC213	4	19	1	3	4	3	1
12CC214	4	21	1	3	4	3	1
12CC215	3	15	1	3	3	3	1
12CC216	4	13	1	3	4	3	1
12CC217	3	8	2	3	4	3	1
12CC218	4	16	1	3	4	3	1
12CC219	1	8	1	3	4	3	1
12CC220	1	10	1	3	1	3	1
12CC221	1	17	1	3	2	3	1
12CC222	4	14	1	3	4	3	2
12CC223	3	19	1	3	4	3	1

12CC224	4	17	1	3	1	3	1
12CC225	4	18	1	3	4	3	1
12CC226	1	14	1	3	2	3	1
12CC227	1	15	2	3	2	3	1
12CC228	4	10	1	3	4	3	1
12CC229	4	12	1	3	4	3	1
12CC230	1	14	1	3	3	3	1
12CC231	4	15	1	3	1	3	1
12CC232	1	9	1	3	2	3	1
12CC233	4	15	1	3	4	3	1
12CC234	1	8	1	3	1	3	1
12CC235	4	7	1	3	4	3	1
12CC236	1	5	2	3	1	2	1
12CC237	4	9	2	3	4	2	1
12CC238	3	12	2	3	3	3	1
12CC239	4	8	2	3	4	3	1
12CC240	1	7	2	3	4	3	1
12CC241	4	22	1	3	4	3	1
12CC242	4	10	2	3	4	3	1
12CC243	2	13	1	3	4	3	1
12CC244	3	14	2	3	1	3	1
12CC245	4	11	1	3	4	3	1
12CC246	4	6	1	3	4	3	1
12CC247	1	10	1	3	2	3	1
12CC248	4	11	1	3	2	3	1
12CC249	4	7	1	3	4	3	1
12CC250	1	7	2	3	1	3	1
12CC251	1	15	2	3	2	3	1
12CC252	4	12	1	3	4	3	1
12CC253	3	11	1	3	3	3	1
12CC254	2	14	1	3	1	3	1
12CC255	1	11	1	3	2	3	1
12CC256	3	19	1	3	2	3	1
12CC257	4	5	2	3	3	3	1
12CC258	1	8	2	3	2	3	1
12CC259	4	6	2	3	4	3	1
12CC260	3	17	2	3	4	3	1
12CC261	2	10	2	3	2	3	1
12CC262	1	14	1	3	1	3	1
12CC263	1	11	2	3	1	3	1
12CC264	4	13	1	3	1	3	1
12CC265	4	9	2	3	3	3	1
12CC266	4	10	2	3	1	3	1
12CC267	3	11	1	3	2	3	1
12CC268	4	13	1	3	4	3	1
12CC269	4	17	1	3	2	3	1

12CC270	4	6	1	3	1	3	1
12CC271	3	10	2	3	1	3	2
12CC272	4	17	1	3	1	2	2
12CC273	4	6	1	3	2	3	1
12CC274	2	18	1	3	4	2	1
12CC275	2	16	1	3	4	3	1
12CC276	2	12	2	3	2	3	1
12CC277	4	15	2	3	1	3	1
12CC278	4	18	1	3	4	3	1
12CC279	2	10	2	3	2	3	1
12CC280	4	16	1	3	4	3	1
12CC281	3	17	1	3	4	3	1
12CC282	2	9	1	3	4	3	1
12CC283	3	18	2	3	2	3	1
12CC284	3	10	1	3	3	3	1
12CC285	4	15	1	3	1	3	1
12CC286	4	12	1	3	4	3	1
12CC287	2	15	1	3	4	3	1
12CC288	4	15	1	3	1	3	1
12CC289	2	16	1	3	4	3	1
12CC290	4	21	1	3	4	3	1
12CC291	4	14	1	3	4	3	1
12CC292	4	15	1	3	4	3	1
12CC293	4	14	2	3	4	3	1
12CC294	1	9	1	3	1	3	1
12CC295	4	7	1	3	4	3	1
12CC296	4	9	1	3	1	3	1
12CC297	2	12	1	3	2	3	1
12CC298	4	8	1	3	1	3	1
12CC299	3	19	1	3	4	3	1
12CC300	3	7	1	3	4	3	1
12CC301	4	13	2	3	1	3	1
12CC302	2	9	1	3	1	3	1
12CC303	3	15	1	3	4	3	1
12CC304	1	8	1	3	1	3	1
12CC305	2	10	1	3	4	3	1
12CC306	4	11	1	3	2	3	1
12CC307	1	7	1	3	1	3	1
12CC308	4	16	1	3	4	3	1
12CC309	4	11	1	3	4	3	1
12CC310	3	8	2	3	2	3	1
12CC311	4	11	1	3	1	3	1
12CC312	4	16	1	3	3	3	1
12CC313	2	20	1	3	2	3	1
12CC314	2	6	1	3	2	3	1
12CC315	3	8	1	3	1	3	1

12CC316	3	6	1	3	1	3	1
12CC317	4	20	1	3	4	3	1
12CC318	4	20	1	3	4	3	1
12CC319	4	13	1	3	4	3	1
12CC320	4	13	1	3	4	3	1
12CC321	4	11	1	3	4	3	1
12CC322	2	16	1	3	1	3	2
12CC323	4	13	2	3	3	3	1
12CC324	4	7	2	3	2	3	1
12CC325	2	14	1	3	1	3	1
12CC326	3	9	1	3	3	3	1
12CC327	4	16	1	3	4	3	1
12CC328	4	14	1	3	3	3	1
12CC329	4	22	1	3	4	3	1
12CC330	4	7	1	3	4	3	1
12CC331	1	16	1	3	1	2	1
12CC332	1	15	1	3	2	2	1
12CC333	2	11	1	3	2	3	1
12CC334	2	10	2	3	1	3	1
12CC335	2	10	2	3	4	3	1
12CC336	2	24	1	3	4	3	1
12CC337	2	15	1	3	1	3	1
12CC338	2	22	2	3	4	3	1
12CC339	4	19	1	3	4	3	1
12CC340	2	15	2	3	2	3	1
12CC341	2	11	1	3	4	3	1
12CC342	2	17	1	3	2	2	1
12CC343	2	18	1	3	2	3	1
12CC344	2	11	2	3	4	3	1
12CC345	2	9	2	3	4	3	1
12CC346	2	17	1	3	4	3	1
12CC347	2	11	2	3	2	3	1
12CC348	2	7	2	1	1	3	1
12CC349	4	17	2	1	4	3	1
12CC350	2	12	1	3	4	3	1
12CC351	4	24	1	3	4	3	1
12CC352	4	23	1	3	4	3	1
12CC353	4	13	1	3	4	3	1
12CC354	4	10	1	3	4	3	1
12CC355	4	19	1	3	4	3	1
12CC356	3	16	1	3	4	3	1
12CC357	1	15	1	3	4	3	1
12CC358	2	10	1	3	4	3	1
12CC359	4	6	2	3	4	3	1
12CC360	4	22	1	3	4	3	1
12CC361	1	9	2	3	4	3	1

12CC362	4	18	1	3	4	3	1
12CC363	3	15	1	3	4	3	1
12CC364	2	11	1	3	4	3	1
12CC365	4	15	1	3	4	3	2
12CC366	4	17	1	3	4	3	1
12CC367	4	18	1	3	4	3	1
12CC368	4	17	1	3	4	3	1
12CC369	4	15	2	3	4	3	1
12CC370	4	12	1	3	4	3	1
12CC371	4	14	1	3	4	3	1
12CC372	4	10	1	3	4	3	1
12CC373	3	17	1	3	4	3	1
12CC374	4	20	1	3	4	3	1
12CC375	4	21	1	3	4	3	1
12CC376	4	19	1	3	4	2	1
12CC377	3	18	1	3	4	3	1
12CC378	4	19	1	3	4	2	1
12CC379	3	10	1	3	4	3	1
12CC380	4	16	1	3	4	3	1
12CC381	2	8	1	3	2	3	1
12CC382	2	16	2	3	1	3	1
12CC383	3	17	2	3	3	3	1
12CC384	1	17	1	3	1	3	1
12CC385	1	14	1	3	3	3	2
12CC386	4	17	1	3	3	3	2
12CC387	4	15	1	3	1	3	1
12CC388	4	19	1	3	4	3	1
12CC389	3	12	1	3	1	3	1
12CC390	4	21	1	3	4	2	1
12CC391	4	14	1	3	4	2	1
12CC392	2	11	1	3	2	3	1
12CC393	4	12	1	3	1	3	1
12CC394	3	19	1	3	4	3	1
12CC395	1	12	1	3	2	3	1
12CC396	4	12	1	3	2	3	1
12CC397	1	16	1	3	3	3	1
12CC398	4	11	1	3	3	3	1
12CC399	3	15	2	3	4	3	1
12CC400	1	18	1	3	4	3	1
12CC401	1	17	1	3	2	3	1
12CC402	2	15	1	3	1	3	1
12CC403	1	10	2	3	1	3	1
12CC404	1	17	1	3	1	3	1
12CC405	4	24	1	3	4	3	1
12CC406	3	16	2	3	1	3	1
12CC407	4	17	1	3	4	3	1

12CC408	4	17	1	3	4	3	1
12CC409	2	19	1	3	4	3	1
12CC410	4	12	2	3	1	3	1
12CC411	4	17	1	3	3	3	1
12CC412	3	8	1	3	1	2	1
12CC413	4	13	1	3	4	3	1
12CC414	4	11	1	3	4	3	1
12CC415	4	16	1	3	4	3	1
12CC416	1	7	2	3	1	3	2
12CC417	1	11	1	3	1	3	1
12CC418	4	9	1	3	2	3	1
12CC419	4	9	1	3	4	3	1
12CC420	3	11	1	3	2	3	1
12CC421	1	7	2	3	2	3	1
12CC422	1	10	2	3	1	3	1
12CC423	1	10	1	3	1	3	1
12CC424	1	12	2	3	1	3	1
12CC425	3	22	1	3	4	3	1
12CC426	1	9	2	3	2	3	1
12CC427	1	13	1	3	3	3	1
12CC428	4	9	2	3	1	3	1
12CC429	1	19	2	3	4	3	1
12CC430	1	10	1	3	1	3	1
12CC431	3	22	1	3	4	3	1
12CC432	4	16	1	3	4	3	1
12CC433	4	15	1	3	4	3	1
12CC434	4	17	2	3	4	3	1
12CC435	4	12	2	3	2	2	1
12CC436	1	14	1	3	2	3	1
12CC437	4	7	2	3	2	3	1
12CC438	1	6	1	3	3	3	1
12CC439	1	15	1	3	2	3	1
12CC440	4	8	2	3	1	3	1
12CC441	1	13	2	3	1	3	1
12CC442	1	10	2	3	1	3	1
12CC443	2	20	2	3	2	3	1
12CC444	1	11	2	3	4	3	1
12CC445	4	13	1	3	1	3	1
12CC446	2	10	1	3	1	3	1
12CC447	1	9	1	3	1	2	1
12CC448	1	21	1	3	1	3	1
12CC449	1	11	2	3	2	3	1
12CC450	1	8	1	3	2	3	2
12CC451	4	7	1	3	4	3	1
12CC452	4	10	1	3	2	3	1
12CC453	4	16	1	3	4	3	1

12CC454	4	9	1	3	4	3	1
12CC455	3	9	1	3	4	3	1
12CC456	4	10	2	3	2	3	1
12CC457	1	9	1	3	1	2	1
12CC458	1	8	1	3	4	3	1
12CC459	2	14	1	3	2	3	1
12CC460	1	8	1	3	1	3	1
12CC461	4	18	1	3	3	3	1
12CC462	1	11	1	3	1	3	1
12CC463	4	6	2	3	2	3	1
12CC464	3	9	2	3	1	3	1
12CC465	4	16	2	3	1	3	1
12CC466	1	8	1	3	1	3	1
12CC467	4	18	1	3	4	3	1
12CC468	4	14	1	3	1	3	1
12CC469	4	14	1	3	4	2	1
12CC470	4	12	1	3	2	3	1
12CC471	3	10	2	3	1	3	1
12CC472	4	15	1	3	3	3	1
12CC473	4	15	1	3	4	3	1
12CC474	4	9	1	3	1	3	1
12CC475	4	8	1	3	2	3	1
12CC476	2	12	1	3	2	3	1
12CC477	4	18	1	3	4	3	1
12CC478	4	16	2	3	2	3	1
12CC479	4	24	1	3	4	2	1
12CC480	4	12	2	3	4	3	2
12CC481	2	6	1	3	2	3	1
12CC482	4	19	1	3	4	3	1
12CC483	3	7	2	3	3	3	1
12CC484	1	11	1	3	4	3	1
12CC485	4	18	1	3	1	3	1
12CC486	1	21	1	3	2	3	1
12CC487	3	6	1	3	1	3	1
12CC488	2	15	1	3	2	3	1
12CC489	4	14	2	3	4	3	1
12CC490	4	17	1	3	2	3	1
12CC491	2	9	1	3	1	3	1
12CC492	4	18	1	3	4	3	1
12CC493	1	8	2	3	1	3	1
12CC494	2	7	1	3	4	1	1
12CC495	4	16	1	3	4	3	1
12CC496	2	10	1	3	1	3	1
12CC497	4	8	1	3	1	3	1
12CC498	4	14	1	3	2	3	1
12CC499	4	8	1	3	1	3	1

12CC500	4	6	1	3	1	3	1
12CC501	4	20	1	3	4	3	1
12CC502	4	15	1	3	4	3	1
12CC503	4	24	1	3	2	3	1
12CC504	4	16	1	3	1	3	1
12CC505	4	26	1	3	4	3	1
12CC506	4	16	1	3	1	3	1
12CC507	4	9	2	3	4	3	1
12CC508	4	5	1	3	1	3	1
12CC509	1	11	2	3	2	3	1
12CC510	2	15	1	3	3	3	1
12CC511	2	10	1	3	2	3	1
12CC512	3	11	1	3	2	3	1
12CC513	1	7	2	3	1	3	1
12CC514	4	15	1	3	3	3	1
12CC515	3	13	1	3	4	3	1
12CC516	4	12	1	3	1	3	1
12CC517	1	8	1	3	1	3	1
12CC518	4	12	2	3	3	3	1
12CC519	1	17	1	3	1	3	1
12CC520	1	12	2	3	2	3	1
12CC521	4	16	2	3	4	3	1
12CC522	4	15	1	3	3	3	1
12CC523	4	18	1	3	2	3	1
12CC524	4	9	1	3	4	3	1
12CC525	1	6	1	3	2	3	1
12CC526	4	13	2	3	4	3	1
12CC527	2	18	1	3	4	3	1
12CC528	4	6	1	3	3	2	1
12CC529	3	9	1	3	1	2	1
12CC530	4	20	1	3	4	3	1
12CC531	2	17	1	3	1	3	1
12CC532	4	18	1	3	4	3	1
12CC533	2	12	1	3	2	3	1
12CC534	4	19	1	3	4	3	2
12CC535	2	11	1	3	1	3	1
12CC536	4	15	1	3	2	3	1
12CC537	4	16	1	3	4	3	1
12CC538	4	5	1	3	4	3	1
12CC539	4	19	1	3	4	3	1
12CC540	4	9	1	3	2	3	1
12CC541	4	7	2	3	1	3	1
12CC542	2	6	1	3	1	3	1
12CC543	1	10	2	3	1	3	1
12CC544	4	15	2	3	4	3	1
12CC545	4	8	2	3	2	3	1

12CC546	1	10	1	3	4	3	1
12CC547	1	9	2	3	1	3	1
12CC548	4	9	1	3	2	3	1
12CC549	4	14	1	3	1	3	1
12CC550	1	12	1	3	1	3	1
12CC551	2	7	2	3	1	3	1
12CC552	1	11	1	3	1	3	1
12CC553	3	7	1	3	2	3	1
12CC554	1	14	1	3	1	3	1
12CC555	3	10	1	3	1	3	1
12CC556	3	9	1	3	1	3	1
12CC557	1	10	1	3	1	3	1
12CC558	1	11	1	3	4	3	1
12CC559	4	13	1	3	4	3	1
12CC560	4	10	1	3	4	3	1
12CC561	4	17	1	3	2	3	1
12CC562	4	8	2	3	1	3	1
12CC563	1	9	1	3	1	3	1
12CC564	2	12	1	3	1	2	1
12CC565	4	20	1	3	4	3	1
12CC566	1	7	1	3	1	3	1
12CC567	2	8	1	3	2	3	1
12CC568	1	12	2	3	1	3	1
12CC569	1	9	2	3	4	3	1
12CC570	1	13	1	3	2	3	1
12CC571	2	11	1	3	1	3	1
12CC572	4	10	1	3	1	3	1
12CC573	4	19	1	3	1	3	1
12CC574	4	11	2	3	1	3	1
12CC575	4	8	1	3	3	3	1
12CC576	2	15	1	3	1	3	1
12CC577	1	12	1	3	4	3	1
12CC578	4	20	1	3	4	3	1
12CC579	4	14	1	3	4	3	1
12CC580	4	12	1	3	1	3	1
12CC581	3	17	1	3	3	3	1
12CC582	4	11	1	3	4	3	1
12CC583	2	10	1	3	4	3	1
12CC584	1	10	1	3	2	3	1
12CC585	4	21	2	3	2	3	1
12CC586	4	22	1	3	4	3	1
12CC587	2	10	2	3	4	3	1
12CC588	3	14	1	3	1	3	1
12CC589	4	16	1	3	2	3	1
12CC590	2	12	1	3	4	3	1
12CC591	4	25	1	3	4	3	1

12CC592	2	14	2	3	1	3	1
12CC593	1	9	1	3	2	3	1
12CC594	1	12	2	3	2	3	2
12CC595	3	6	1	3	1	3	1
12CC596	2	10	1	3	1	3	1
12CC597	2	13	1	3	4	3	1
12CC598	2	13	1	3	4	3	1
12CC599	2	14	2	3	4	3	1
12CC600	2	19	1	3	2	2	1
12CC601	2	11	1	3	4	3	1
12CC602	2	13	2	3	4	3	1
12CC603	2	13	2	3	4	3	1
12CC604	2	9	1	3	4	3	1
12CC605	2	7	2	3	3	3	1
12CC606	2	8	2	3	2	3	1
12CC607	2	9	1	3	1	3	1
12CC608	2	10	1	3	1	3	1
12CC609	2	6	1	3	4	3	1
12CC610	2	6	2	3	4	3	1
12CC611	2	13	1	3	4	3	1
12CC612	2	16	1	3	1	3	1
12CC613	3	13	1	3	4	3	1
12CC614	2	20	1	3	1	3	1
12CC615	2	9	1	3	2	3	1
12CC616	4	21	1	3	1	3	1
12CC617	1	8	1	3	1	3	1
12CC618	1	15	1	3	1	3	1
12CC619	1	12	2	3	1	3	1
12CC620	1	12	1	3	1	3	1
12CC621	2	16	1	3	1	3	1
12CC622	2	10	1	3	1	3	1
12CC623	2	11	1	3	1	3	1
12CC624	3	17	1	3	1	3	1
12CC625	1	10	1	3	1	3	1
12CC626	1	5	1	3	1	3	1
12CC627	4	7	1	3	1	3	1
12CC628	2	11	2	3	1	3	1
12CC629	1	17	1	3	1	2	1
12CC630	4	13	1	3	1	3	1
12CC631	1	10	1	3	1	3	1
12CC632	4	9	2	3	1	3	1
12CC633	1	9	1	3	1	3	1
12CC634	1	14	2	3	1	3	1
12CC635	2	11	1	3	1	3	1
12CC636	4	22	1	3	1	3	1
12CC637	2	6	1	3	1	3	1

12CC638	1	12	1	3	2	2	1
12CC639	4	16	1	3	2	3	1
12CC640	4	24	1	3	2	3	2
12CC641	1	8	1	3	2	3	1
12CC642	2	10	1	3	2	3	1
12CC643	2	13	1	3	2	3	1
12CC644	3	10	1	3	2	3	1
12CC645	1	8	2	3	2	3	1
12CC646	1	12	2	3	2	3	1
12CC647	1	5	1	3	2	3	1
12CC648	4	17	1	3	2	3	1
12CC649	1	13	2	3	2	3	1
12CC650	1	12	2	3	2	3	1
12CC651	2	16	1	3	2	3	1
12CC652	4	15	1	3	2	3	1
12CC653	4	12	2	3	2	3	1
12CC654	4	14	1	3	2	3	2
12CC655	4	7	2	3	3	3	2
12CC656	4	15	1	3	3	3	1
12CC657	4	16	1	3	3	3	1
12CC658	3	30	1	3	3	3	1
12CC659	4	9	1	3	3	3	1
12CC660	2	18	1	3	3	3	1
12CC661	1	9	1	3	3	3	1
12CC662	1	5	1	3	3	3	1
12CC663	3	18	1	3	3	3	1
12CC664	2	10	2	3	3	3	1
12CC665	2	12	1	3	4	3	1
12CC666	4	17	1	3	4	3	2
12CC667	1	18	2	3	4	3	1
12CC668	4	21	1	3	4	3	1
12CC669	1	8	1	3	4	3	1
12CC670	3	9	1	3	4	3	1
12CC671	1	11	2	3	4	3	1
12CC672	4	17	2	3	4	3	1
12CC673	4	16	1	3	4	3	1
12CC674	2	16	1	3	4	3	1
12CC675	4	22	1	3	4	3	1
12CC676	4	6	1	3	4	3	1
12CC677	1	11	1	3	4	3	1
12CC678	4	18	1	3	4	3	1
12CC679	4	18	1	3	4	3	1
12CC680	4	13	2	3	4	3	1
12CC681	4	17	1	3	4	3	1
12CC682	4	27	1	3	4	3	1
12CC683	2	8	1	3	4	3	1

12CC684	4	18	1	3	4	3	1
12CC685	1	20	1	3	4	3	1
12CC686	1	14	1	3	4	3	1
12CC687	4	12	1	3	4	3	1
12CC688	4	23	1	3	4	3	1
12CC689	4	13	2	3	4	3	1
12CC690	2	9	1	3	4	3	1
12CC691	4	13	1	3	4	3	1
12CC692	4	21	1	3	4	3	1
12CC693	4	10	1	3	4	3	1
12CC694	2	15	1	3	4	3	1
12CC695	2	16	1	3	4	3	1
12CC696	4	27	1	3	4	3	1
12CC697	4	19	1	3	4	3	1
12CC698	4	15	1	3	4	3	1
12CC699	4	18	1	3	4	3	1
12CC700	4	19	2	3	4	3	1
12CC701	4	17	1	3	4	3	1
12CC702	4	20	1	3	4	3	1
12CC703	4	16	1	3	4	3	1
12CC704	4	18	1	3	4	3	1
12CC705	4	14	1	3	4	3	1
12CC706	4	16	2	3	4	2	1
12CC707	4	11	1	3	4	2	1
12CC708	4	13	1	3	4	3	1
12CC709	4	14	1	3	4	3	1
12CC710	3	19	1	3	4	3	1
12CC711	4	21	1	3	4	3	1
12CC712	3	11	2	3	4	3	1
12CC713	4	17	1	3	4	3	1
12CC714	1	8	1	3	4	3	1
12CC715	3	13	1	3	4	3	1
12CC716	4	17	1	3	4	3	1
12CC717	4	11	2	3	4	3	1
12CC718	4	11	1	3	4	3	2
12CC719	1	7	2	3	4	3	1
12CC720	2	9	1	3	4	3	1
12CC721	2	10	1	3	4	3	1
12CC722	1	13	1	3	2	3	1
12CC723	2	14	1	3	2	3	1
12CC724	4	21	2	3	2	3	1
12CC725	1	12	1	3	2	3	1
12CC726	3	8	1	3	2	2	1
12CC727	1	7	2	3	2	3	1
12CC728	4	16	1	3	2	3	1
12CC729	3	17	1	3	2	3	1

12CC730	3	17	1	3	2	3	1
12CC731	1	18	1	3	2	3	1
12CC732	2	9	1	3	2	3	1
12CC733	4	10	1	3	1	3	1
12CC734	1	8	2	3	1	3	1
12CC735	2	9	2	3	1	3	1
12CC736	1	12	2	3	1	3	1
12CC737	2	23	2	3	1	2	1
12CC738	2	10	2	3	1	3	1
12CC739	1	13	1	3	1	3	1
12CC740	1	10	1	3	1	3	1
12CC741	1	12	1	3	1	3	1
12CC742	2	15	1	3	1	3	1
12CC743	1	21	1	3	1	3	1
12CC744	4	25	1	3	4	3	1
12CC745	4	20	1	3	4	3	1
12CC746	3	7	1	3	4	3	1
12CC747	4	20	1	3	4	3	1
12CC748	4	14	1	3	4	3	1
12CC749	4	9	1	3	4	3	2
12CC750	4	20	1	3	4	3	2
12CC751	1	11	2	3	4	3	1
12CC752	4	25	2	3	4	3	1
12CC753	3	11	1	3	4	3	1
12CC754	4	13	1	3	4	2	1
12CC755	4	12	1	3	4	3	1
12CC756	4	15	1	3	4	3	1
12CC757	4	25	1	3	4	3	1
12CC758	4	14	1	3	4	3	1
12CC759	1	15	1	3	4	3	1
12CC760	4	26	1	3	4	3	1
12CC761	4	20	1	3	4	3	1
12CC762	4	10	1	3	4	3	1
12CC763	1	20	2	3	4	3	1
12CC764	4	16	1	3	4	3	1
12CC765	2	8	1	3	4	3	1
12CC766	4	20	1	3	4	3	1
12CC767	4	25	1	3	4	3	1
12CC768	3	18	1	3	4	3	1
12CC769	3	16	1	3	4	3	1
12CC770	1	8	1	3	4	3	1
12CC771	4	21	1	3	4	3	1
12CC772	4	15	1	3	4	3	1
12CC773	3	9	1	3	4	3	1
12CC774	4	19	1	3	4	3	1
12CC775	4	15	2	3	4	3	1

12CC776	1	9	2	3	4	3	1
12CC777	4	10	1	3	4	3	1
12CC778	4	16	1	3	4	3	1
12CC779	4	26	1	3	4	3	1
12CC780	1	8	2	3	4	3	1
12CC781	1	14	1	3	4	3	1
12CC782	4	22	2	3	4	3	1
12CC783	2	19	1	3	4	3	1
12CC784	4	11	1	3	4	3	1
12CC785	2	10	2	3	4	3	1
12CC786	3	10	1	3	4	3	1
12CC787	3	15	2	3	4	3	1
12CC788	4	14	1	3	4	3	1
12CC789	4	18	1	3	4	3	1
12CC790	1	10	2	3	4	3	1
12CC791	4	10	1	3	4	3	1
12CC792	2	11	1	3	4	3	1
12CC793	4	15	2	3	4	3	1
12CC794	2	10	2	3	4	2	1
12CC795	2	12	1	3	4	2	1
12CC796	2	9	1	3	4	3	1
12CC797	3	12	2	3	3	3	1
12CC798	2	8	1	3	3	3	1
12CC799	4	15	1	3	3	3	1
12CC800	4	22	1	3	3	3	1
12CC801	3	10	1	3	3	3	1
12CC802	4	23	1	3	3	3	2
12CC803	1	8	1	3	3	3	1
12CC804	4	13	1	3	3	3	1
12CC805	1	14	1	3	3	2	1
12CC806	2	14	2	3	3	3	1
12CC807	4	11	1	3	3	3	1
12CC808	3	18	2	3	3	3	1
12CC809	1	11	2	3	2	3	1
12CC810	1	11	1	3	2	3	1
12CC811	1	10	1	3	2	3	1
12CC812	1	15	1	3	2	3	1
12CC813	1	6	1	3	2	2	1
12CC814	4	14	1	3	2	3	1
12CC815	2	7	1	3	2	3	1
12CC816	3	11	1	3	2	3	1
12CC817	1	12	2	3	2	3	1
12CC818	4	8	2	3	2	3	1
12CC819	3	8	1	3	2	3	1
12CC820	1	8	1	3	2	3	1
12CC821	1	16	1	3	2	3	1

12CC822	4	21	1	3	2	2	1
12CC823	2	13	1	3	2	3	1
12CC824	4	13	2	3	2	3	1
12CC825	3	8	1	3	2	3	1
12CC826	2	17	1	3	2	3	1
12CC827	3	10	1	3	2	3	1
12CC828	4	16	2	3	2	3	1
12CC829	4	17	2	3	2	3	1
12CC830	1	9	1	3	2	3	1
12CC831	2	13	2	3	2	3	1
12CC832	2	19	2	3	2	3	1
12CC833	4	9	1	3	2	3	1
12CC834	1	9	2	3	2	3	1
12CC835	3	12	1	3	2	2	1
12CC836	1	8	1	3	2	2	1
12CC837	1	12	1	3	2	3	1
12CC838	3	19	1	3	2	3	1
12CC839	1	14	1	3	2	3	1
12CC840	2	12	1	3	2	3	1
12CC841	4	9	1	3	2	3	1
12CC842	2	15	1	3	2	3	1
12CC843	1	9	1	3	1	3	1
12CC844	4	10	1	3	1	2	1
12CC845	1	9	1	3	1	3	1
12CC846	3	8	1	3	1	3	1
12CC847	1	15	1	3	1	3	1
12CC848	4	18	2	3	1	3	1
12CC849	4	13	1	3	1	3	1
12CC850	1	14	2	3	1	3	1
12CC851	1	14	1	3	1	3	2
12CC852	4	5	1	3	1	3	1
12CC853	1	10	1	3	1	3	1
12CC854	1	10	1	3	1	3	1
12CC855	1	13	1	3	1	3	1
12CC856	1	13	1	3	1	3	1
12CC857	1	6	2	3	1	2	1
12CC858	3	9	1	3	1	3	1
12CC859	3	7	1	3	1	3	1
12CC860	4	11	2	3	1	3	1
12CC861	1	6	1	3	1	3	1
12CC862	4	16	1	3	1	3	1
12CC863	2	12	1	3	1	2	1
12CC864	4	13	1	3	1	2	1
12CC865	1	13	1	3	1	3	1
12CC866	1	14	1	3	1	3	1
12CC867	4	12	2	3	1	3	1

12CC868	1	18	1	3	1	3	1
12CC869	4	7	1	3	1	3	1
12CC870	3	8	1	3	1	3	1
12CC871	4	13	1	3	1	3	1
12CC872	1	14	1	3	1	3	1
12CC873	1	8	1	3	1	2	1
12CC874	1	11	1	3	1	3	1
12CC875	4	9	1	3	1	3	1
12CC876	1	13	1	3	1	3	1
12CC877	2	17	2	3	1	3	1
12CC878	2	8	2	3	1	3	1
12CC879	4	9	1	3	1	3	1
12CC880	3	10	2	3	1	3	1
12CC881	2	10	1	3	1	2	1
12CC882	2	5	2	3	1	3	1
12CC883	2	6	2	3	1	3	1
12CC884	4	18	1	3	4	3	1
12CC885	2	11	1	3	1	3	1
12CC886	4	13	1	3	4	3	1
12CC887	2	10	1	3	4	2	1
12CC888	3	14	2	3	1	3	1
12CC889	4	23	1	3	4	2	1
12CC890	3	10	1	3	4	3	1
12CC891	4	19	1	3	4	3	1
12CC892	4	17	1	3	2	3	1
12CC893	2	12	1	3	1	3	2
12CC894	2	23	2	3	2	3	1
12CC895	3	12	1	3	4	3	1
12CC896	4	21	1	1	4	3	1
12CC897	4	13	2	3	4	1	1
12CC898	4	20	1	3	4	3	1
12CC899	3	19	1	3	4	3	1
12CC900	4	17	1	3	3	3	1
12CC901	1	12	1	3	4	3	1
12CC902	1	12	2	3	4	3	1
12CC903	2	7	2	3	1	3	1
12CC904	1	14	1	3	2	3	1
12CC905	3	18	1	3	2	3	1
12CC906	1	7	1	3	1	3	1
12CC907	2	18	1	3	2	3	1
12CC908	1	13	2	3	1	3	1
12CC909	2	21	1	3	2	3	1
12CC910	3	16	1	3	1	3	1
12CC911	4	11	1	3	4	3	1
12CC912	4	13	2	3	4	3	1
12CC913	4	18	1	3	4	2	1

12CC914	4	15	1	3	4	2	1
12CC915	1	10	2	3	1	3	1
12CC916	4	19	1	3	4	3	1
12CC917	3	11	1	3	4	3	1
12CC918	1	7	1	3	1	3	1
12CC919	1	13	1	3	2	3	1
12CC920	4	27	1	3	4	3	1
12CC921	2	8	2	3	1	3	1
12CC922	3	14	2	3	4	3	1
12CC923	4	21	1	3	4	3	1
12CC924	3	15	1	3	4	3	1
12CC925	4	21	1	3	3	2	1
12CC926	3	22	1	3	4	3	1
12CC927	3	21	1	3	4	3	1
12CC928	2	24	1	3	4	3	1
12CC929	1	9	1	3	2	3	1
12CC930	1	13	1	3	2	3	1
12CC931	2	20	1	3	4	3	1
12CC932	4	16	1	3	4	3	1
12CC933	4	11	1	3	1	3	1
12CC934	4	16	1	3	1	3	1
12CC935	4	13	1	3	1	3	2
12CC936	2	15	1	3	4	2	1
12CC937	2	17	1	3	1	3	1
12CC938	4	16	2	3	2	3	1
12CC939	4	22	1	3	4	3	1
12CC940	4	8	1	3	3	3	1
12CC941	3	16	2	3	1	2	1
12CC942	2	9	1	3	3	3	1
12CC943	4	21	1	3	4	3	1
12CC944	4	19	1	3	4	3	1
12CC945	2	8	2	3	4	3	1
12CC946	4	22	1	3	4	2	1
12CC947	4	23	1	3	4	3	1
12CC948	4	17	1	3	4	3	1
12CC949	4	28	1	3	4	3	1
12CC950	3	17	1	3	4	3	1
12CC951	1	9	1	3	2	3	1
12CC952	4	8	1	3	1	3	1
12CC953	1	11	1	3	1	3	1
12CC954	3	20	1	3	2	3	1
12CC955	4	26	1	3	2	3	1
12CC956	1	6	2	1	1	3	1
12CC957	3	26	1	3	4	3	1
12CC958	1	19	1	3	3	3	1
12CC959	1	10	1	3	2	3	1

12CC960	3	16	1	3	4	3	1
12CC961	4	17	1	3	4	2	1
12CC962	4	18	1	3	1	3	1
12CC963	3	11	1	3	2	3	1
12CC964	4	12	2	3	4	3	1
12CC965	4	18	1	3	4	3	2
12CC966	4	13	1	3	4	3	1
12CC967	4	17	2	3	4	3	1
12CC968	4	24	1	3	4	3	1
12CC969	2	19	1	3	4	3	1
12CC970	3	7	1	3	2	3	1
12CC971	1	8	2	3	1	3	1
12CC972	1	12	2	3	2	3	1
12CC973	4	19	1	3	4	3	1
12CC974	2	9	1	3	1	3	1
12CC975	4	14	1	3	4	3	2
12CC976	4	8	1	3	4	3	2
12CC977	2	15	2	3	4	3	1
12CC978	3	11	1	3	2	3	1
12CC979	4	19	1	3	4	2	1
12CC980	4	13	1	3	4	2	1
12CC981	2	22	1	3	4	3	1
12CC982	2	15	1	3	4	3	1
12CC983	2	10	2	3	4	2	1
12CC984	2	18	1	3	4	3	1
12CC985	2	19	1	3	4	3	1
12CC986	2	19	2	3	2	3	1
12CC987	3	12	1	3	2	2	1
12CC988	4	14	1	3	4	3	1
12CC989	3	15	2	3	4	3	1
12CC990	1	18	1	3	2	3	1
12CC991	4	19	1	3	4	3	1
12CC992	2	9	1	3	4	3	1
12CC993	4	17	1	3	2	3	1
12CC994	3	17	1	3	4	3	1
12CC995	4	16	1	3	2	2	1
12CC996	4	18	2	3	4	3	1
12CC997	4	12	1	3	2	3	1
12CC998	1	12	1	3	2	3	1
12CC999	4	12	2	3	2	3	1
12CC1000	2	10	1	3	2	3	1
12CC1001	3	9	2	3	2	3	1
12CC1002	2	14	1	3	4	3	1
12CC1003	3	11	1	3	1	3	1
12CC1004	1	15	1	3	1	2	2
12CC1005	3	9	2	3	2	3	2

12CC1006	1	10	2	3	1	3	1
12CC1007	1	10	2	3	2	3	1
12CC1008	2	11	1	3	2	3	1
12CC1009	2	6	2	3	1	3	1
12CC1010	4	6	1	3	2	3	1
12CC1011	1	13	1	3	1	3	1
12CC1012	1	9	1	3	1	1	1
12CC1013	4	7	1	3	1	3	1
12CC1014	4	6	1	3	1	3	1
12CC1015	2	6	1	3	1	3	1
12CC1016	3	10	1	3	1	3	1
12CC1017	2	16	1	3	2	3	1
12CC1018	4	12	2	3	2	3	1
12CC1019	1	15	2	3	1	3	1
12CC1020	4	9	1	3	1	2	1
12CC1021	4	12	1	3	2	3	1
12CC1022	4	9	1	3	2	3	1
12CC1023	4	14	1	3	1	3	1
12CC1024	4	15	1	3	2	3	1
12CC1025	4	13	2	3	2	3	1
12CC1026	4	18	1	3	2	3	1
12CC1027	1	9	2	3	1	2	1
12CC1028	4	12	2	3	2	3	1
12CC1029	4	18	1	3	2	3	1
12CC1030	1	16	1	3	1	3	1
12CC1031	1	8	2	3	1	3	1
12CC1032	4	16	1	3	2	3	1
12CC1033	4	16	1	3	2	3	1
12CC1034	4	18	1	3	2	3	1
12CC1035	3	17	1	3	2	3	1
12CC1036	4	13	2	3	2	3	1
12CC1037	4	12	1	3	2	3	1
12CC1038	4	22	1	3	2	3	1
12CC1039	3	21	1	3	2	2	1
12CC1040	1	10	2	3	1	2	2
12CC1041	2	13	1	3	1	3	1
12CC1042	1	11	1	3	1	3	2
12CC1043	2	7	1	3	2	3	2
12CC1044	1	14	2	3	1	3	2
12CC1045	4	10	1	3	1	3	2
12CC1046	1	12	1	3	2	3	1
12CC1047	4	22	1	3	2	2	1
12CC1048	1	7	1	3	2	3	1
12CC1049	1	12	1	3	1	3	1
12CC1050	4	15	1	3	1	3	1
12CC1051	4	22	1	3	2	3	1

12CC1052	4	10	2	3	2	3	1
12CC1053	3	13	2	3	2	3	1
12CC1054	2	7	1	3	2	3	1
12CC1055	3	11	1	3	2	3	1
12CC1056	1	10	2	3	1	3	1
12CC1057	3	14	1	3	1	3	1
12CC1058	1	13	2	3	1	3	1
12CC1059	2	22	1	3	2	2	1
12CC1060	4	17	1	3	2	3	1
12CC1061	1	11	1	3	1	3	1
12CC1062	4	9	1	3	2	2	1
12CC1063	1	14	2	3	2	3	1
12CC1064	4	12	2	3	1	3	1
12CC1065	2	21	1	3	1	3	1
12CC1066	1	7	1	3	2	3	1
12CC1067	1	13	1	3	2	3	1
12CC1068	4	10	1	3	2	2	1
12CC1069	3	11	1	3	2	3	1
12CC1070	1	16	1	3	1	3	1
12CC1071	1	13	2	3	2	3	1
12CC1072	4	13	2	3	2	3	1
12CC1073	4	12	2	3	2	2	1
12CC1074	4	11	1	3	2	3	1
12CC1075	1	11	1	3	2	3	1
12CC1076	4	19	1	3	2	2	1
12CC1077	4	20	1	3	2	3	1
12CC1078	4	15	1	3	2	3	1
12CC1079	4	10	1	3	2	3	1
12CC1080	4	21	1	3	2	3	1
12CC1081	2	9	1	3	2	3	1
12CC1082	1	10	1	3	2	3	1
12CC1083	4	28	1	3	2	3	1
12CC1084	4	21	1	3	2	3	1
12CC1085	2	20	1	3	2	3	1
12CC1086	4	17	1	3	2	1	1
12CC1087	2	8	1	3	1	3	2
12CC1088	4	11	1	3	2	3	1
12CC1089	1	11	2	3	2	3	1
12CC1090	4	26	1	3	2	3	1
12CC1091	1	7	2	3	1	3	1
12CC1092	2	12	1	3	2	3	1
12CC1093	1	19	2	3	2	3	1
12CC1094	4	15	2	3	2	3	1
12CC1095	1	10	2	3	2	3	1
12CC1096	1	12	2	3	2	2	1
12CC1097	4	19	1	3	2	2	1

12CC1098	1	13	1	3	2	3	1
12CC1099	4	17	1	3	2	3	1
12CC1100	1	5	1	3	2	3	1
12CC1101	3	9	1	3	2	3	1
12CC1102	1	6	1	3	1	3	1
12CC1103	3	9	1	3	2	3	1
12CC1104	1	9	1	3	1	3	1
12CC1105	1	15	2	3	1	2	1
12CC1106	1	10	2	3	2	3	1
12CC1107	4	24	1	3	2	3	1
12CC1108	2	8	1	3	1	3	1
12CC1109	4	14	1	3	2	3	1
12CC1110	4	11	1	3	1	3	1
12CC1111	2	14	1	3	2	3	1
12CC1112	1	9	1	3	2	3	1
12CC1113	1	8	1	3	2	2	1
12CC1114	4	16	2	3	2	3	1
12CC1115	1	16	1	3	1	3	1
12CC1116	4	17	2	3	2	3	1
12CC1117	4	21	1	3	2	3	1
12CC1118	4	15	1	3	2	3	1
12CC1119	1	14	2	3	2	3	1
12CC1120	1	12	1	3	2	3	1
12CC1121	3	22	1	3	2	3	1
12CC1122	1	14	1	3	2	3	1
12CC1123	1	9	2	3	1	3	1
12CC1124	3	10	1	3	1	2	2
12CC1125	1	11	1	3	2	3	1
12CC1126	4	8	2	3	2	3	1
12CC1127	1	7	2	3	1	3	1
12CC1128	4	11	1	3	2	2	1
12CC1129	1	5	1	3	1	3	1
12CC1130	4	18	1	3	2	3	1
12CC1131	4	6	1	3	2	3	1
12CC1132	1	10	1	3	1	3	1
12CC1133	1	10	1	3	1	3	1
12CC1134	3	9	2	3	1	3	1
12CC1135	1	15	2	3	2	3	1
12CC1136	4	9	2	3	2	3	1
12CC1137	4	9	2	3	1	2	1
12CC1138	1	12	1	3	2	3	1
12CC1139	4	8	1	3	2	3	1
12CC1140	3	15	1	3	2	3	1
12CC1141	4	24	1	3	2	3	1
12CC1142	4	17	1	3	1	3	1
12CC1143	3	15	1	3	2	3	1

12CC1144	4	16	1	3	2	3	1
12CC1145	4	14	1	3	2	3	1
12CC1146	4	27	1	3	2	3	1
12CC1147	4	16	1	3	1	2	1
12CC1148	4	9	1	3	2	2	1
12CC1149	2	12	1	3	2	3	1
12CC1150	4	14	1	3	2	3	1
12CC1151	3	20	1	3	1	3	1
12CC1152	4	12	2	3	1	3	1
12CC1153	4	15	1	3	2	2	1
12CC1154	1	7	1	3	2	3	1
12CC1155	4	21	1	3	2	3	1
12CC1156	4	24	1	3	2	3	1
12CC1157	4	8	2	3	2	3	1
12CC1158	4	20	1	3	2	2	1
12CC1159	4	12	2	3	2	3	1
12CC1160	1	14	2	3	1	3	1
12CC1161	4	9	1	3	2	3	1
12CC1162	1	5	1	3	2	3	1
12CC1163	3	23	1	3	2	3	1
12CC1164	3	10	2	3	2	2	1
12CC1165	4	12	1	3	2	3	2
12CC1166	1	14	1	3	1	3	1
12CC1167	4	17	1	3	1	3	1
12CC1168	4	11	1	3	2	3	1
12CC1169	1	13	1	3	1	3	1
12CC1170	2	8	1	3	2	2	1
12CC1171	1	12	1	3	2	3	1
12CC1172	3	9	1	3	2	3	1
12CC1173	2	10	1	3	2	3	1
12CC1174	4	13	1	3	2	3	1
12CC1175	4	19	1	3	2	3	1
12CC1176	1	9	1	3	1	2	1
12CC1177	1	14	1	3	2	3	1
12CC1178	3	8	1	3	1	3	1
12CC1179	2	11	2	3	1	3	1
12CC1180	4	9	2	3	2	3	1
12CC1181	3	9	1	3	1	3	1
12CC1182	1	22	1	3	2	2	1
12CC1183	1	5	1	3	1	3	1
12CC1184	1	14	1	3	1	2	1
12CC1185	4	12	2	3	2	3	1
12CC1186	1	9	2	3	2	3	1
12CC1187	1	6	1	3	1	3	1
12CC1188	1	17	1	3	2	3	1
12CC1189	4	5	1	3	1	2	2

12CC1190	1	15	2	3	2	3	1
12CC1191	3	14	1	3	2	3	1
12CC1192	1	11	1	3	1	3	1
12CC1193	3	18	1	3	2	2	1
12CC1194	1	14	1	3	2	3	1
12CC1195	1	14	1	3	2	3	1
12CC1196	4	6	1	3	2	3	1
12CC1197	4	18	2	3	2	3	1
12CC1198	1	14	2	3	1	3	1
12CC1199	1	14	2	3	1	2	2
12CC1200	1	9	2	3	1	3	1
12CC1201	4	12	1	2	4	3	1

APPENDIX E

OUTPUTS BASED ON THE FIRST DATASET

S/N	REGISTRATION NO.	SCORE ACHIEVED	RISK STATUS	PERFORMANCE PREDICTED
1	12CC001	52	High risk	Average Performance
2	12CC002	52	High risk	Average Performance
3	12CC003	58	High risk	Average Performance
4	12CC004	74	Risk free	Excellent Performance
5	12CC005	66	Low risk	Good Performance
6	12CC006	80	Risk free	Excellent Performance
7	12CC007	48	Very high risk	Poor Performance
8	12CC008	32	Very high risk	Poor Performance
9	12CC009	56	High risk	Average Performance
10	12CC010	62	Low risk	Good Performance
11	12CC011	48	Very high risk	Poor Performance
12	12CC012	54	High risk	Average Performance
13	12CC013	70	Risk free	Excellent Performance
14	12CC014	62	Low risk	Good Performance
15	12CC015	64	Low risk	Good Performance
16	12CC016	40	Very high risk	Poor Performance
17	12CC017	74	Risk free	Excellent Performance
18	12CC018	42	Very high risk	Poor Performance
19	12CC019	54	High risk	Average Performance
20	12CC020	74	Risk free	Excellent Performance
21	12CC021	36	Very high risk	Poor Performance
22	12CC022	50	High risk	Average Performance
23	12CC023	52	High risk	Average Performance
24	12CC024	40	Very high risk	Poor Performance
25	12CC025	34	Very high risk	Poor Performance
26	12CC026	46	Very high risk	Poor Performance
27	12CC027	86	Risk free	Excellent Performance
28	12CC028	76	Risk free	Excellent Performance
29	12CC029	48	Very high risk	Poor Performance
30	12CC030	56	High risk	Average Performance
31	12CC031	48	Very high risk	Poor Performance
32	12CC032	66	Low risk	Good Performance
33	12CC033	48	Very high risk	Poor Performance
34	12CC034	40	Very high risk	Poor Performance
35	12CC035	38	Very high risk	Poor Performance

36	12CC036	48	Very high risk	Poor Performance
37	12CC037	76	Risk free	Excellent Performance
38	12CC038	50	High risk	Average Performance
39	12CC039	54	High risk	Average Performance
40	12CC040	60	Low risk	Good Performance
41	12CC041	70	Risk free	Excellent Performance
42	12CC042	46	Very high risk	Poor Performance
43	12CC043	60	Low risk	Good Performance
44	12CC044	62	Low risk	Good Performance
45	12CC045	44	Very high risk	Poor Performance
46	12CC046	74	Risk free	Excellent Performance
47	12CC047	56	High risk	Average Performance
48	12CC048	54	High risk	Average Performance
49	12CC049	80	Risk free	Excellent Performance
50	12CC050	58	High risk	Average Performance
51	12CC051	78	Risk free	Excellent Performance
52	12CC052	40	Very high risk	Poor Performance
53	12CC053	46	Very high risk	Poor Performance
54	12CC054	72	Risk free	Excellent Performance
55	12CC055	48	Very high risk	Poor Performance
56	12CC056	64	Low risk	Good Performance
57	12CC057	74	Risk free	Excellent Performance
58	12CC058	64	Low risk	Good Performance
59	12CC059	36	Very high risk	Poor Performance
60	12CC060	36	Very high risk	Poor Performance
61	12CC061	44	Very high risk	Poor Performance
62	12CC062	52	High risk	Average Performance
63	12CC063	50	High risk	Average Performance
64	12CC064	64	Low risk	Good Performance
65	12CC065	46	Very high risk	Poor Performance
66	12CC066	40	Very high risk	Poor Performance
67	12CC067	86	Risk free	Excellent Performance
68	12CC068	76	Risk free	Excellent Performance
69	12CC069	42	Very high risk	Poor Performance
70	12CC070	48	Very high risk	Poor Performance
71	12CC071	54	High risk	Average Performance
72	12CC072	52	High risk	Average Performance
73	12CC073	40	Very high risk	Poor Performance
74	12CC074	70	Risk free	Excellent Performance
75	12CC075	62	Low risk	Good Performance
76	12CC076	84	Risk free	Excellent Performance
77	12CC077	56	High risk	Average Performance
78	12CC078	44	Very high risk	Poor Performance
79	12CC079	46	Very high risk	Poor Performance
80	12CC080	46	Very high risk	Poor Performance

81	12CC081	52	High risk	Average Performance
82	12CC082	48	Very high risk	Poor Performance
83	12CC083	52	High risk	Average Performance
84	12CC084	84	Risk free	Excellent Performance
85	12CC085	42	Very high risk	Poor Performance
86	12CC086	32	Very high risk	Poor Performance
87	12CC087	56	High risk	Average Performance
88	12CC088	52	High risk	Average Performance
89	12CC089	50	High risk	Average Performance
90	12CC090	46	Very high risk	Poor Performance
91	12CC091	56	High risk	Average Performance
92	12CC092	58	High risk	Average Performance
93	12CC093	36	Very high risk	Poor Performance
94	12CC094	60	Low risk	Good Performance
95	12CC095	66	Low risk	Good Performance
96	12CC096	52	High risk	Average Performance
97	12CC097	50	High risk	Average Performance
98	12CC098	46	Very high risk	Poor Performance
99	12CC099	68	Low risk	Good Performance
100	12CC100	66	Low risk	Good Performance
101	12CC1000	44	Very high risk	Poor Performance
102	12CC1001	46	Very high risk	Poor Performance
103	12CC1002	56	High risk	Average Performance
104	12CC1003	46	Very high risk	Poor Performance
105	12CC1004	50	High risk	Average Performance
106	12CC1005	48	Very high risk	Poor Performance
107	12CC1006	42	Very high risk	Poor Performance
108	12CC1007	44	Very high risk	Poor Performance
109	12CC1008	46	Very high risk	Poor Performance
110	12CC1009	36	Very high risk	Poor Performance
111	12CC101	48	Very high risk	Poor Performance
112	12CC1010	40	Very high risk	Poor Performance
113	12CC1011	46	Very high risk	Poor Performance
114	12CC1012	34	Very high risk	Poor Performance
115	12CC1013	40	Very high risk	Poor Performance
116	12CC1014	38	Very high risk	Poor Performance
117	12CC1015	34	Very high risk	Poor Performance
118	12CC1016	44	Very high risk	Poor Performance
119	12CC1017	56	High risk	Average Performance
120	12CC1018	54	High risk	Average Performance
121	12CC1019	52	High risk	Average Performance
122	12CC102	50	High risk	Average Performance
123	12CC1020	42	Very high risk	Poor Performance
124	12CC1021	52	High risk	Average Performance
125	12CC1022	46	Very high risk	Poor Performance

126	12CC1023	54	High risk	Average Performance
127	12CC1024	58	High risk	Average Performance
128	12CC1025	56	High risk	Average Performance
129	12CC1026	64	Low risk	Good Performance
130	12CC1027	38	Very high risk	Poor Performance
131	12CC1028	54	High risk	Average Performance
132	12CC1029	64	Low risk	Good Performance
133	12CC103	56	High risk	Average Performance
134	12CC1030	52	High risk	Average Performance
135	12CC1031	38	Very high risk	Poor Performance
136	12CC1032	60	Low risk	Good Performance
137	12CC1033	60	Low risk	Good Performance
138	12CC1034	64	Low risk	Good Performance
139	12CC1035	60	Low risk	Good Performance
140	12CC1036	56	High risk	Average Performance
141	12CC1037	52	High risk	Average Performance
142	12CC1038	72	Risk free	Excellent Performance
143	12CC1039	66	Low risk	Good Performance
144	12CC104	40	Very high risk	Poor Performance
145	12CC1040	42	Very high risk	Poor Performance
146	12CC1041	48	Very high risk	Poor Performance
147	12CC1042	44	Very high risk	Poor Performance
148	12CC1043	40	Very high risk	Poor Performance
149	12CC1044	52	High risk	Average Performance
150	12CC1045	48	Very high risk	Poor Performance
151	12CC1046	46	Very high risk	Poor Performance
152	12CC1047	70	Risk free	Excellent Performance
153	12CC1048	36	Very high risk	Poor Performance
154	12CC1049	44	Very high risk	Poor Performance
155	12CC105	42	Very high risk	Poor Performance
156	12CC1050	56	High risk	Average Performance
157	12CC1051	72	Risk free	Excellent Performance
158	12CC1052	50	High risk	Average Performance
159	12CC1053	54	High risk	Average Performance
160	12CC1054	38	Very high risk	Poor Performance
161	12CC1055	48	Very high risk	Poor Performance
162	12CC1056	42	Very high risk	Poor Performance
163	12CC1057	52	High risk	Average Performance
164	12CC1058	48	Very high risk	Poor Performance
165	12CC1059	66	Low risk	Good Performance
166	12CC106	48	Very high risk	Poor Performance
167	12CC1060	62	Low risk	Good Performance
168	12CC1061	42	Very high risk	Poor Performance
169	12CC1062	44	Very high risk	Poor Performance
170	12CC1063	52	High risk	Average Performance

171	12CC1064	52	High risk	Average Performance
172	12CC1065	64	Low risk	Good Performance
173	12CC1066	36	Very high risk	Poor Performance
174	12CC1067	48	Very high risk	Poor Performance
175	12CC1068	46	Very high risk	Poor Performance
176	12CC1069	48	Very high risk	Poor Performance
177	12CC107	44	Very high risk	Poor Performance
178	12CC1070	52	High risk	Average Performance
179	12CC1071	50	High risk	Average Performance
180	12CC1072	56	High risk	Average Performance
181	12CC1073	52	High risk	Average Performance
182	12CC1074	50	High risk	Average Performance
183	12CC1075	44	Very high risk	Poor Performance
184	12CC1076	64	Low risk	Good Performance
185	12CC1077	68	Low risk	Good Performance
186	12CC1078	58	High risk	Average Performance
187	12CC1079	48	Very high risk	Poor Performance
188	12CC108	46	Very high risk	Poor Performance
189	12CC1080	70	Risk free	Excellent Performance
190	12CC1081	42	Very high risk	Poor Performance
191	12CC1082	42	Very high risk	Poor Performance
192	12CC1083	84	Risk free	Excellent Performance
193	12CC1084	70	Risk free	Excellent Performance
194	12CC1085	64	Low risk	Good Performance
195	12CC1086	58	High risk	Average Performance
196	12CC1087	40	Very high risk	Poor Performance
197	12CC1088	50	High risk	Average Performance
198	12CC1089	46	Very high risk	Poor Performance
199	12CC109	46	Very high risk	Poor Performance
200	12CC1090	80	Risk free	Excellent Performance
201	12CC1091	36	Very high risk	Poor Performance
202	12CC1092	48	Very high risk	Poor Performance
203	12CC1093	62	Low risk	Good Performance
204	12CC1094	60	Low risk	Good Performance
205	12CC1095	44	Very high risk	Poor Performance
206	12CC1096	46	Very high risk	Poor Performance
207	12CC1097	64	Low risk	Good Performance
208	12CC1098	48	Very high risk	Poor Performance
209	12CC1099	62	Low risk	Good Performance
210	12CC110	54	High risk	Average Performance
211	12CC1100	32	Very high risk	Poor Performance
212	12CC1101	44	Very high risk	Poor Performance
213	12CC1102	32	Very high risk	Poor Performance
214	12CC1103	44	Very high risk	Poor Performance
215	12CC1104	38	Very high risk	Poor Performance

216	12CC1105	50	High risk	Average Performance
217	12CC1106	44	Very high risk	Poor Performance
218	12CC1107	76	Risk free	Excellent Performance
219	12CC1108	38	Very high risk	Poor Performance
220	12CC1109	56	High risk	Average Performance
221	12CC111	58	High risk	Average Performance
222	12CC1110	48	Very high risk	Poor Performance
223	12CC1111	52	High risk	Average Performance
224	12CC1112	40	Very high risk	Poor Performance
225	12CC1113	36	Very high risk	Poor Performance
226	12CC1114	62	Low risk	Good Performance
227	12CC1115	52	High risk	Average Performance
228	12CC1116	64	Low risk	Good Performance
229	12CC1117	70	Risk free	Excellent Performance
230	12CC1118	58	High risk	Average Performance
231	12CC1119	52	High risk	Average Performance
232	12CC112	52	High risk	Average Performance
233	12CC1120	46	Very high risk	Poor Performance
234	12CC1121	70	Risk free	Excellent Performance
235	12CC1122	50	High risk	Average Performance
236	12CC1123	40	Very high risk	Poor Performance
237	12CC1124	44	Very high risk	Poor Performance
238	12CC1125	44	Very high risk	Poor Performance
239	12CC1126	46	Very high risk	Poor Performance
240	12CC1127	36	Very high risk	Poor Performance
241	12CC1128	48	Very high risk	Poor Performance
242	12CC1129	30	Very high risk	Poor Performance
243	12CC113	60	Low risk	Good Performance
244	12CC1130	64	Low risk	Good Performance
245	12CC1131	40	Very high risk	Poor Performance
246	12CC1132	40	Very high risk	Poor Performance
247	12CC1133	40	Very high risk	Poor Performance
248	12CC1134	44	Very high risk	Poor Performance
249	12CC1135	54	High risk	Average Performance
250	12CC1136	48	Very high risk	Poor Performance
251	12CC1137	44	Very high risk	Poor Performance
252	12CC1138	46	Very high risk	Poor Performance
253	12CC1139	44	Very high risk	Poor Performance
254	12CC114	58	High risk	Average Performance
255	12CC1140	56	High risk	Average Performance
256	12CC1141	76	Risk free	Excellent Performance
257	12CC1142	60	Low risk	Good Performance
258	12CC1143	56	High risk	Average Performance
259	12CC1144	60	Low risk	Good Performance
260	12CC1145	56	High risk	Average Performance

261	12CC1146	82	Risk free	Excellent Performance
262	12CC1147	56	High risk	Average Performance
263	12CC1148	44	Very high risk	Poor Performance
264	12CC1149	48	Very high risk	Poor Performance
265	12CC115	72	Risk free	Excellent Performance
266	12CC1150	56	High risk	Average Performance
267	12CC1151	64	Low risk	Good Performance
268	12CC1152	52	High risk	Average Performance
269	12CC1153	56	High risk	Average Performance
270	12CC1154	36	Very high risk	Poor Performance
271	12CC1155	70	Risk free	Excellent Performance
272	12CC1156	76	Risk free	Excellent Performance
273	12CC1157	46	Very high risk	Poor Performance
274	12CC1158	66	Low risk	Good Performance
275	12CC1159	54	High risk	Average Performance
276	12CC116	46	Very high risk	Poor Performance
277	12CC1160	50	High risk	Average Performance
278	12CC1161	46	Very high risk	Poor Performance
279	12CC1162	32	Very high risk	Poor Performance
280	12CC1163	72	Risk free	Excellent Performance
281	12CC1164	46	Very high risk	Poor Performance
282	12CC1165	54	High risk	Average Performance
283	12CC1166	48	Very high risk	Poor Performance
284	12CC1167	60	Low risk	Good Performance
285	12CC1168	50	High risk	Average Performance
286	12CC1169	46	Very high risk	Poor Performance
287	12CC117	60	Low risk	Good Performance
288	12CC1170	38	Very high risk	Poor Performance
289	12CC1171	46	Very high risk	Poor Performance
290	12CC1172	44	Very high risk	Poor Performance
291	12CC1173	44	Very high risk	Poor Performance
292	12CC1174	54	High risk	Average Performance
293	12CC1175	66	Low risk	Good Performance
294	12CC1176	36	Very high risk	Poor Performance
295	12CC1177	50	High risk	Average Performance
296	12CC1178	40	Very high risk	Poor Performance
297	12CC1179	46	Very high risk	Poor Performance
298	12CC118	70	Risk free	Excellent Performance
299	12CC1180	48	Very high risk	Poor Performance
300	12CC1181	42	Very high risk	Poor Performance
301	12CC1182	64	Low risk	Good Performance
302	12CC1183	30	Very high risk	Poor Performance
303	12CC1184	46	Very high risk	Poor Performance
304	12CC1185	54	High risk	Average Performance
305	12CC1186	42	Very high risk	Poor Performance

306	12CC1187	32	Very high risk	Poor Performance
307	12CC1188	56	High risk	Average Performance
308	12CC1189	36	Very high risk	Poor Performance
309	12CC119	42	Very high risk	Poor Performance
310	12CC1190	54	High risk	Average Performance
311	12CC1191	54	High risk	Average Performance
312	12CC1192	42	Very high risk	Poor Performance
313	12CC1193	60	Low risk	Good Performance
314	12CC1194	50	High risk	Average Performance
315	12CC1195	50	High risk	Average Performance
316	12CC1196	40	Very high risk	Poor Performance
317	12CC1197	66	Low risk	Good Performance
318	12CC1198	50	High risk	Average Performance
319	12CC1199	50	High risk	Average Performance
320	12CC120	58	High risk	Average Performance
321	12CC1200	40	Very high risk	Poor Performance
322	12CC121	66	Low risk	Good Performance
323	12CC122	74	Risk free	Excellent Performance
324	12CC123	52	High risk	Average Performance
325	12CC124	42	Very high risk	Poor Performance
326	12CC125	62	Low risk	Good Performance
327	12CC126	48	Very high risk	Poor Performance
328	12CC127	58	High risk	Average Performance
329	12CC128	64	Low risk	Good Performance
330	12CC129	38	Very high risk	Poor Performance
331	12CC130	56	High risk	Average Performance
332	12CC131	54	High risk	Average Performance
333	12CC132	50	High risk	Average Performance
334	12CC133	54	High risk	Average Performance
335	12CC134	68	Low risk	Good Performance
336	12CC135	38	Very high risk	Poor Performance
337	12CC136	40	Very high risk	Poor Performance
338	12CC137	52	High risk	Average Performance
339	12CC138	56	High risk	Average Performance
340	12CC139	42	Very high risk	Poor Performance
341	12CC140	52	High risk	Average Performance
342	12CC141	66	Low risk	Good Performance
343	12CC142	76	Risk free	Excellent Performance
344	12CC143	72	Risk free	Excellent Performance
345	12CC144	44	Very high risk	Poor Performance
346	12CC145	64	Low risk	Good Performance
347	12CC146	80	Risk free	Excellent Performance
348	12CC147	76	Risk free	Excellent Performance
349	12CC148	82	Risk free	Excellent Performance
350	12CC149	54	High risk	Average Performance

351	12CC150	68	Low risk	Good Performance
352	12CC151	70	Risk free	Excellent Performance
353	12CC152	66	Low risk	Good Performance
354	12CC153	52	High risk	Average Performance
355	12CC154	58	High risk	Average Performance
356	12CC155	72	Risk free	Excellent Performance
357	12CC156	42	Very high risk	Poor Performance
358	12CC157	34	Very high risk	Poor Performance
359	12CC158	56	High risk	Average Performance
360	12CC159	64	Low risk	Good Performance
361	12CC160	60	Low risk	Good Performance
362	12CC161	52	High risk	Average Performance
363	12CC162	52	High risk	Average Performance
364	12CC163	50	High risk	Average Performance
365	12CC164	62	Low risk	Good Performance
366	12CC165	48	Very high risk	Poor Performance
367	12CC166	62	Low risk	Good Performance
368	12CC167	50	High risk	Average Performance
369	12CC168	60	Low risk	Good Performance
370	12CC169	42	Very high risk	Poor Performance
371	12CC170	44	Very high risk	Poor Performance
372	12CC171	38	Very high risk	Poor Performance
373	12CC172	46	Very high risk	Poor Performance
374	12CC173	60	Low risk	Good Performance
375	12CC174	58	High risk	Average Performance
376	12CC175	60	Low risk	Good Performance
377	12CC176	76	Risk free	Excellent Performance
378	12CC177	42	Very high risk	Poor Performance
379	12CC178	50	High risk	Average Performance
380	12CC179	54	High risk	Average Performance
381	12CC180	72	Risk free	Excellent Performance
382	12CC181	46	Very high risk	Poor Performance
383	12CC182	52	High risk	Average Performance
384	12CC183	44	Very high risk	Poor Performance
385	12CC184	58	High risk	Average Performance
386	12CC185	44	Very high risk	Poor Performance
387	12CC186	64	Low risk	Good Performance
388	12CC187	62	Low risk	Good Performance
389	12CC188	68	Low risk	Good Performance
390	12CC189	56	High risk	Average Performance
391	12CC190	72	Risk free	Excellent Performance
392	12CC191	68	Low risk	Good Performance
393	12CC192	64	Low risk	Good Performance
394	12CC193	70	Risk free	Excellent Performance
395	12CC194	68	Low risk	Good Performance

396	12CC195	54	High risk	Average Performance
397	12CC196	58	High risk	Average Performance
398	12CC197	72	Risk free	Excellent Performance
399	12CC198	70	Risk free	Excellent Performance
400	12CC199	38	Very high risk	Poor Performance
401	12CC200	42	Very high risk	Poor Performance
402	12CC201	74	Risk free	Excellent Performance
403	12CC202	54	High risk	Average Performance
404	12CC203	50	High risk	Average Performance
405	12CC204	56	High risk	Average Performance
406	12CC205	62	Low risk	Good Performance
407	12CC206	64	Low risk	Good Performance
408	12CC207	44	Very high risk	Poor Performance
409	12CC208	38	Very high risk	Poor Performance
410	12CC209	46	Very high risk	Poor Performance
411	12CC210	58	High risk	Average Performance
412	12CC211	58	High risk	Average Performance
413	12CC212	56	High risk	Average Performance
414	12CC213	70	Risk free	Excellent Performance
415	12CC214	74	Risk free	Excellent Performance
416	12CC215	58	High risk	Average Performance
417	12CC216	58	High risk	Average Performance
418	12CC217	48	Very high risk	Poor Performance
419	12CC218	64	Low risk	Good Performance
420	12CC219	42	Very high risk	Poor Performance
421	12CC220	40	Very high risk	Poor Performance
422	12CC221	56	High risk	Average Performance
423	12CC222	62	Low risk	Good Performance
424	12CC223	68	Low risk	Good Performance
425	12CC224	60	Low risk	Good Performance
426	12CC225	68	Low risk	Good Performance
427	12CC226	50	High risk	Average Performance
428	12CC227	54	High risk	Average Performance
429	12CC228	52	High risk	Average Performance
430	12CC229	56	High risk	Average Performance
431	12CC230	52	High risk	Average Performance
432	12CC231	56	High risk	Average Performance
433	12CC232	40	Very high risk	Poor Performance
434	12CC233	62	Low risk	Good Performance
435	12CC234	36	Very high risk	Poor Performance
436	12CC235	46	Very high risk	Poor Performance
437	12CC236	30	Very high risk	Poor Performance
438	12CC237	50	High risk	Average Performance
439	12CC238	54	High risk	Average Performance
440	12CC239	50	High risk	Average Performance

441	12CC240	42	Very high risk	Poor Performance
442	12CC241	76	Risk free	Excellent Performance
443	12CC242	54	High risk	Average Performance
444	12CC243	54	High risk	Average Performance
445	12CC244	54	High risk	Average Performance
446	12CC245	54	High risk	Average Performance
447	12CC246	44	Very high risk	Poor Performance
448	12CC247	42	Very high risk	Poor Performance
449	12CC248	50	High risk	Average Performance
450	12CC249	46	Very high risk	Poor Performance
451	12CC250	36	Very high risk	Poor Performance
452	12CC251	54	High risk	Average Performance
453	12CC252	56	High risk	Average Performance
454	12CC253	50	High risk	Average Performance
455	12CC254	50	High risk	Average Performance
456	12CC255	44	Very high risk	Poor Performance
457	12CC256	64	Low risk	Good Performance
458	12CC257	42	Very high risk	Poor Performance
459	12CC258	40	Very high risk	Poor Performance
460	12CC259	46	Very high risk	Poor Performance
461	12CC260	66	Low risk	Good Performance
462	12CC261	46	Very high risk	Poor Performance
463	12CC262	48	Very high risk	Poor Performance
464	12CC263	44	Very high risk	Poor Performance
465	12CC264	52	High risk	Average Performance
466	12CC265	50	High risk	Average Performance
467	12CC266	48	Very high risk	Poor Performance
468	12CC267	48	Very high risk	Poor Performance
469	12CC268	58	High risk	Average Performance
470	12CC269	62	Low risk	Good Performance
471	12CC270	38	Very high risk	Poor Performance
472	12CC271	48	Very high risk	Poor Performance
473	12CC272	60	Low risk	Good Performance
474	12CC273	40	Very high risk	Poor Performance
475	12CC274	62	Low risk	Good Performance
476	12CC275	60	Low risk	Good Performance
477	12CC276	50	High risk	Average Performance
478	12CC277	58	High risk	Average Performance
479	12CC278	68	Low risk	Good Performance
480	12CC279	46	Very high risk	Poor Performance
481	12CC280	64	Low risk	Good Performance
482	12CC281	64	Low risk	Good Performance
483	12CC282	46	Very high risk	Poor Performance
484	12CC283	64	Low risk	Good Performance
485	12CC284	48	Very high risk	Poor Performance

486	12CC285	56	High risk	Average Performance
487	12CC286	56	High risk	Average Performance
488	12CC287	58	High risk	Average Performance
489	12CC288	56	High risk	Average Performance
490	12CC289	60	Low risk	Good Performance
491	12CC290	74	Risk free	Excellent Performance
492	12CC291	60	Low risk	Good Performance
493	12CC292	62	Low risk	Good Performance
494	12CC293	62	Low risk	Good Performance
495	12CC294	38	Very high risk	Poor Performance
496	12CC295	46	Very high risk	Poor Performance
497	12CC296	44	Very high risk	Poor Performance
498	12CC297	48	Very high risk	Poor Performance
499	12CC298	42	Very high risk	Poor Performance
500	12CC299	68	Low risk	Good Performance
501	12CC300	44	Very high risk	Poor Performance
502	12CC301	54	High risk	Average Performance
503	12CC302	40	Very high risk	Poor Performance
504	12CC303	60	Low risk	Good Performance
505	12CC304	36	Very high risk	Poor Performance
506	12CC305	48	Very high risk	Poor Performance
507	12CC306	50	High risk	Average Performance
508	12CC307	34	Very high risk	Poor Performance
509	12CC308	64	Low risk	Good Performance
510	12CC309	54	High risk	Average Performance
511	12CC310	44	Very high risk	Poor Performance
512	12CC311	48	Very high risk	Poor Performance
513	12CC312	62	Low risk	Good Performance
514	12CC313	64	Low risk	Good Performance
515	12CC314	36	Very high risk	Poor Performance
516	12CC315	40	Very high risk	Poor Performance
517	12CC316	36	Very high risk	Poor Performance
518	12CC317	72	Risk free	Excellent Performance
519	12CC318	72	Risk free	Excellent Performance
520	12CC319	58	High risk	Average Performance
521	12CC320	58	High risk	Average Performance
522	12CC321	54	High risk	Average Performance
523	12CC322	56	High risk	Average Performance
524	12CC323	58	High risk	Average Performance
525	12CC324	44	Very high risk	Poor Performance
526	12CC325	50	High risk	Average Performance
527	12CC326	46	Very high risk	Poor Performance
528	12CC327	64	Low risk	Good Performance
529	12CC328	58	High risk	Average Performance
530	12CC329	76	Risk free	Excellent Performance

531	12CC330	46	Very high risk	Poor Performance
532	12CC331	50	High risk	Average Performance
533	12CC332	50	High risk	Average Performance
534	12CC333	46	Very high risk	Poor Performance
535	12CC334	44	Very high risk	Poor Performance
536	12CC335	50	High risk	Average Performance
537	12CC336	76	Risk free	Excellent Performance
538	12CC337	52	High risk	Average Performance
539	12CC338	74	Risk free	Excellent Performance
540	12CC339	70	Risk free	Excellent Performance
541	12CC340	56	High risk	Average Performance
542	12CC341	50	High risk	Average Performance
543	12CC342	56	High risk	Average Performance
544	12CC343	60	Low risk	Good Performance
545	12CC344	52	High risk	Average Performance
546	12CC345	48	Very high risk	Poor Performance
547	12CC346	62	Low risk	Good Performance
548	12CC347	48	Very high risk	Poor Performance
549	12CC348	34	Very high risk	Poor Performance
550	12CC349	64	Low risk	Good Performance
551	12CC350	52	High risk	Average Performance
552	12CC351	80	Risk free	Excellent Performance
553	12CC352	78	Risk free	Excellent Performance
554	12CC353	58	High risk	Average Performance
555	12CC354	52	High risk	Average Performance
556	12CC355	70	Risk free	Excellent Performance
557	12CC356	62	Low risk	Good Performance
558	12CC357	56	High risk	Average Performance
559	12CC358	48	Very high risk	Poor Performance
560	12CC359	46	Very high risk	Poor Performance
561	12CC360	76	Risk free	Excellent Performance
562	12CC361	46	Very high risk	Poor Performance
563	12CC362	68	Low risk	Good Performance
564	12CC363	60	Low risk	Good Performance
565	12CC364	50	High risk	Average Performance
566	12CC365	64	Low risk	Good Performance
567	12CC366	66	Low risk	Good Performance
568	12CC367	68	Low risk	Good Performance
569	12CC368	66	Low risk	Good Performance
570	12CC369	64	Low risk	Good Performance
571	12CC370	56	High risk	Average Performance
572	12CC371	60	Low risk	Good Performance
573	12CC372	52	High risk	Average Performance
574	12CC373	64	Low risk	Good Performance
575	12CC374	72	Risk free	Excellent Performance

576	12CC375	74	Risk free	Excellent Performance
577	12CC376	68	Low risk	Good Performance
578	12CC377	66	Low risk	Good Performance
579	12CC378	68	Low risk	Good Performance
580	12CC379	50	High risk	Average Performance
581	12CC380	64	Low risk	Good Performance
582	12CC381	40	Very high risk	Poor Performance
583	12CC382	56	High risk	Average Performance
584	12CC383	64	Low risk	Good Performance
585	12CC384	54	High risk	Average Performance
586	12CC385	54	High risk	Average Performance
587	12CC386	66	Low risk	Good Performance
588	12CC387	56	High risk	Average Performance
589	12CC388	70	Risk free	Excellent Performance
590	12CC389	48	Very high risk	Poor Performance
591	12CC390	72	Risk free	Excellent Performance
592	12CC391	58	High risk	Average Performance
593	12CC392	46	Very high risk	Poor Performance
594	12CC393	50	High risk	Average Performance
595	12CC394	68	Low risk	Good Performance
596	12CC395	46	Very high risk	Poor Performance
597	12CC396	52	High risk	Average Performance
598	12CC397	56	High risk	Average Performance
599	12CC398	52	High risk	Average Performance
600	12CC399	62	Low risk	Good Performance
601	12CC400	62	Low risk	Good Performance
602	12CC401	56	High risk	Average Performance
603	12CC402	52	High risk	Average Performance
604	12CC403	42	Very high risk	Poor Performance
605	12CC404	54	High risk	Average Performance
606	12CC405	80	Risk free	Excellent Performance
607	12CC406	58	High risk	Average Performance
608	12CC407	66	Low risk	Good Performance
609	12CC408	66	Low risk	Good Performance
610	12CC409	66	Low risk	Good Performance
611	12CC410	52	High risk	Average Performance
612	12CC411	64	Low risk	Good Performance
613	12CC412	38	Very high risk	Poor Performance
614	12CC413	58	High risk	Average Performance
615	12CC414	54	High risk	Average Performance
616	12CC415	64	Low risk	Good Performance
617	12CC416	38	Very high risk	Poor Performance
618	12CC417	42	Very high risk	Poor Performance
619	12CC418	46	Very high risk	Poor Performance
620	12CC419	50	High risk	Average Performance

621	12CC420	48	Very high risk	Poor Performance
622	12CC421	38	Very high risk	Poor Performance
623	12CC422	42	Very high risk	Poor Performance
624	12CC423	40	Very high risk	Poor Performance
625	12CC424	46	Very high risk	Poor Performance
626	12CC425	74	Risk free	Excellent Performance
627	12CC426	42	Very high risk	Poor Performance
628	12CC427	50	High risk	Average Performance
629	12CC428	46	Very high risk	Poor Performance
630	12CC429	66	Low risk	Good Performance
631	12CC430	40	Very high risk	Poor Performance
632	12CC431	74	Risk free	Excellent Performance
633	12CC432	64	Low risk	Good Performance
634	12CC433	62	Low risk	Good Performance
635	12CC434	68	Low risk	Good Performance
636	12CC435	52	High risk	Average Performance
637	12CC436	50	High risk	Average Performance
638	12CC437	44	Very high risk	Poor Performance
639	12CC438	36	Very high risk	Poor Performance
640	12CC439	52	High risk	Average Performance
641	12CC440	44	Very high risk	Poor Performance
642	12CC441	48	Very high risk	Poor Performance
643	12CC442	42	Very high risk	Poor Performance
644	12CC443	66	Low risk	Good Performance
645	12CC444	50	High risk	Average Performance
646	12CC445	52	High risk	Average Performance
647	12CC446	42	Very high risk	Poor Performance
648	12CC447	36	Very high risk	Poor Performance
649	12CC448	62	Low risk	Good Performance
650	12CC449	46	Very high risk	Poor Performance
651	12CC450	40	Very high risk	Poor Performance
652	12CC451	46	Very high risk	Poor Performance
653	12CC452	48	Very high risk	Poor Performance
654	12CC453	64	Low risk	Good Performance
655	12CC454	50	High risk	Average Performance
656	12CC455	48	Very high risk	Poor Performance
657	12CC456	50	High risk	Average Performance
658	12CC457	36	Very high risk	Poor Performance
659	12CC458	42	Very high risk	Poor Performance
660	12CC459	52	High risk	Average Performance
661	12CC460	36	Very high risk	Poor Performance
662	12CC461	66	Low risk	Good Performance
663	12CC462	42	Very high risk	Poor Performance
664	12CC463	42	Very high risk	Poor Performance
665	12CC464	44	Very high risk	Poor Performance

666	12CC465	60	Low risk	Good Performance
667	12CC466	36	Very high risk	Poor Performance
668	12CC467	68	Low risk	Good Performance
669	12CC468	54	High risk	Average Performance
670	12CC469	58	High risk	Average Performance
671	12CC470	52	High risk	Average Performance
672	12CC471	46	Very high risk	Poor Performance
673	12CC472	60	Low risk	Good Performance
674	12CC473	62	Low risk	Good Performance
675	12CC474	44	Very high risk	Poor Performance
676	12CC475	44	Very high risk	Poor Performance
677	12CC476	48	Very high risk	Poor Performance
678	12CC477	68	Low risk	Good Performance
679	12CC478	62	Low risk	Good Performance
680	12CC479	78	Risk free	Excellent Performance
681	12CC480	60	Low risk	Good Performance
682	12CC481	36	Very high risk	Poor Performance
683	12CC482	70	Risk free	Excellent Performance
684	12CC483	44	Very high risk	Poor Performance
685	12CC484	48	Very high risk	Poor Performance
686	12CC485	62	Low risk	Good Performance
687	12CC486	64	Low risk	Good Performance
688	12CC487	36	Very high risk	Poor Performance
689	12CC488	54	High risk	Average Performance
690	12CC489	62	Low risk	Good Performance
691	12CC490	62	Low risk	Good Performance
692	12CC491	40	Very high risk	Poor Performance
693	12CC492	68	Low risk	Good Performance
694	12CC493	38	Very high risk	Poor Performance
695	12CC494	38	Very high risk	Poor Performance
696	12CC495	64	Low risk	Good Performance
697	12CC496	42	Very high risk	Poor Performance
698	12CC497	42	Very high risk	Poor Performance
699	12CC498	56	High risk	Average Performance
700	12CC499	42	Very high risk	Poor Performance
701	12CC500	38	Very high risk	Poor Performance
702	12CC501	72	Risk free	Excellent Performance
703	12CC502	62	Low risk	Good Performance
704	12CC503	76	Risk free	Excellent Performance
705	12CC504	58	High risk	Average Performance
706	12CC505	84	Risk free	Excellent Performance
707	12CC506	58	High risk	Average Performance
708	12CC507	52	High risk	Average Performance
709	12CC508	36	Very high risk	Poor Performance
710	12CC509	46	Very high risk	Poor Performance

711	12CC510	56	High risk	Average Performance
712	12CC511	44	Very high risk	Poor Performance
713	12CC512	48	Very high risk	Poor Performance
714	12CC513	36	Very high risk	Poor Performance
715	12CC514	60	Low risk	Good Performance
716	12CC515	56	High risk	Average Performance
717	12CC516	50	High risk	Average Performance
718	12CC517	36	Very high risk	Poor Performance
719	12CC518	56	High risk	Average Performance
720	12CC519	54	High risk	Average Performance
721	12CC520	48	Very high risk	Poor Performance
722	12CC521	66	Low risk	Good Performance
723	12CC522	60	Low risk	Good Performance
724	12CC523	64	Low risk	Good Performance
725	12CC524	50	High risk	Average Performance
726	12CC525	34	Very high risk	Poor Performance
727	12CC526	60	Low risk	Good Performance
728	12CC527	64	Low risk	Good Performance
729	12CC528	40	Very high risk	Poor Performance
730	12CC529	40	Very high risk	Poor Performance
731	12CC530	72	Risk free	Excellent Performance
732	12CC531	56	High risk	Average Performance
733	12CC532	68	Low risk	Good Performance
734	12CC533	48	Very high risk	Poor Performance
735	12CC534	72	Risk free	Excellent Performance
736	12CC535	44	Very high risk	Poor Performance
737	12CC536	58	High risk	Average Performance
738	12CC537	64	Low risk	Good Performance
739	12CC538	42	Very high risk	Poor Performance
740	12CC539	70	Risk free	Excellent Performance
741	12CC540	46	Very high risk	Poor Performance
742	12CC541	42	Very high risk	Poor Performance
743	12CC542	34	Very high risk	Poor Performance
744	12CC543	42	Very high risk	Poor Performance
745	12CC544	64	Low risk	Good Performance
746	12CC545	46	Very high risk	Poor Performance
747	12CC546	46	Very high risk	Poor Performance
748	12CC547	40	Very high risk	Poor Performance
749	12CC548	46	Very high risk	Poor Performance
750	12CC549	54	High risk	Average Performance
751	12CC550	44	Very high risk	Poor Performance
752	12CC551	38	Very high risk	Poor Performance
753	12CC552	42	Very high risk	Poor Performance
754	12CC553	40	Very high risk	Poor Performance
755	12CC554	48	Very high risk	Poor Performance

756	12CC555	44	Very high risk	Poor Performance
757	12CC556	42	Very high risk	Poor Performance
758	12CC557	40	Very high risk	Poor Performance
759	12CC558	48	Very high risk	Poor Performance
760	12CC559	58	High risk	Average Performance
761	12CC560	52	High risk	Average Performance
762	12CC561	62	Low risk	Good Performance
763	12CC562	44	Very high risk	Poor Performance
764	12CC563	38	Very high risk	Poor Performance
765	12CC564	44	Very high risk	Poor Performance
766	12CC565	72	Risk free	Excellent Performance
767	12CC566	34	Very high risk	Poor Performance
768	12CC567	40	Very high risk	Poor Performance
769	12CC568	46	Very high risk	Poor Performance
770	12CC569	46	Very high risk	Poor Performance
771	12CC570	48	Very high risk	Poor Performance
772	12CC571	44	Very high risk	Poor Performance
773	12CC572	46	Very high risk	Poor Performance
774	12CC573	64	Low risk	Good Performance
775	12CC574	50	High risk	Average Performance
776	12CC575	46	Very high risk	Poor Performance
777	12CC576	52	High risk	Average Performance
778	12CC577	50	High risk	Average Performance
779	12CC578	72	Risk free	Excellent Performance
780	12CC579	60	Low risk	Good Performance
781	12CC580	50	High risk	Average Performance
782	12CC581	62	Low risk	Good Performance
783	12CC582	54	High risk	Average Performance
784	12CC583	48	Very high risk	Poor Performance
785	12CC584	42	Very high risk	Poor Performance
786	12CC585	72	Risk free	Excellent Performance
787	12CC586	76	Risk free	Excellent Performance
788	12CC587	50	High risk	Average Performance
789	12CC588	52	High risk	Average Performance
790	12CC589	60	Low risk	Good Performance
791	12CC590	52	High risk	Average Performance
792	12CC591	82	Risk free	Excellent Performance
793	12CC592	52	High risk	Average Performance
794	12CC593	40	Very high risk	Poor Performance
795	12CC594	50	High risk	Average Performance
796	12CC595	36	Very high risk	Poor Performance
797	12CC596	42	Very high risk	Poor Performance
798	12CC597	54	High risk	Average Performance
799	12CC598	54	High risk	Average Performance
800	12CC599	58	High risk	Average Performance

801	12CC600	60	Low risk	Good Performance
802	12CC601	50	High risk	Average Performance
803	12CC602	56	High risk	Average Performance
804	12CC603	56	High risk	Average Performance
805	12CC604	46	Very high risk	Poor Performance
806	12CC605	42	Very high risk	Poor Performance
807	12CC606	42	Very high risk	Poor Performance
808	12CC607	40	Very high risk	Poor Performance
809	12CC608	42	Very high risk	Poor Performance
810	12CC609	40	Very high risk	Poor Performance
811	12CC610	42	Very high risk	Poor Performance
812	12CC611	54	High risk	Average Performance
813	12CC612	54	High risk	Average Performance
814	12CC613	56	High risk	Average Performance
815	12CC614	62	Low risk	Good Performance
816	12CC615	42	Very high risk	Poor Performance
817	12CC616	68	Low risk	Good Performance
818	12CC617	36	Very high risk	Poor Performance
819	12CC618	50	High risk	Average Performance
820	12CC619	46	Very high risk	Poor Performance
821	12CC620	44	Very high risk	Poor Performance
822	12CC621	54	High risk	Average Performance
823	12CC622	42	Very high risk	Poor Performance
824	12CC623	44	Very high risk	Poor Performance
825	12CC624	58	High risk	Average Performance
826	12CC625	40	Very high risk	Poor Performance
827	12CC626	30	Very high risk	Poor Performance
828	12CC627	40	Very high risk	Poor Performance
829	12CC628	46	Very high risk	Poor Performance
830	12CC629	52	High risk	Average Performance
831	12CC630	52	High risk	Average Performance
832	12CC631	40	Very high risk	Poor Performance
833	12CC632	46	Very high risk	Poor Performance
834	12CC633	38	Very high risk	Poor Performance
835	12CC634	50	High risk	Average Performance
836	12CC635	44	Very high risk	Poor Performance
837	12CC636	70	Risk free	Excellent Performance
838	12CC637	34	Very high risk	Poor Performance
839	12CC638	44	Very high risk	Poor Performance
840	12CC639	60	Low risk	Good Performance
841	12CC640	78	Risk free	Excellent Performance
842	12CC641	38	Very high risk	Poor Performance
843	12CC642	44	Very high risk	Poor Performance
844	12CC643	50	High risk	Average Performance
845	12CC644	46	Very high risk	Poor Performance

846	12CC645	40	Very high risk	Poor Performance
847	12CC646	48	Very high risk	Poor Performance
848	12CC647	32	Very high risk	Poor Performance
849	12CC648	62	Low risk	Good Performance
850	12CC649	50	High risk	Average Performance
851	12CC650	48	Very high risk	Poor Performance
852	12CC651	56	High risk	Average Performance
853	12CC652	58	High risk	Average Performance
854	12CC653	54	High risk	Average Performance
855	12CC654	58	High risk	Average Performance
856	12CC655	48	Very high risk	Poor Performance
857	12CC656	60	Low risk	Good Performance
858	12CC657	62	Low risk	Good Performance
859	12CC658	88	Risk free	Excellent Performance
860	12CC659	48	Very high risk	Poor Performance
861	12CC660	62	Low risk	Good Performance
862	12CC661	42	Very high risk	Poor Performance
863	12CC662	34	Very high risk	Poor Performance
864	12CC663	64	Low risk	Good Performance
865	12CC664	48	Very high risk	Poor Performance
866	12CC665	52	High risk	Average Performance
867	12CC666	68	Low risk	Good Performance
868	12CC667	64	Low risk	Good Performance
869	12CC668	74	Risk free	Excellent Performance
870	12CC669	42	Very high risk	Poor Performance
871	12CC670	48	Very high risk	Poor Performance
872	12CC671	50	High risk	Average Performance
873	12CC672	68	Low risk	Good Performance
874	12CC673	64	Low risk	Good Performance
875	12CC674	60	Low risk	Good Performance
876	12CC675	76	Risk free	Excellent Performance
877	12CC676	44	Very high risk	Poor Performance
878	12CC677	48	Very high risk	Poor Performance
879	12CC678	68	Low risk	Good Performance
880	12CC679	68	Low risk	Good Performance
881	12CC680	60	Low risk	Good Performance
882	12CC681	66	Low risk	Good Performance
883	12CC682	86	Risk free	Excellent Performance
884	12CC683	44	Very high risk	Poor Performance
885	12CC684	68	Low risk	Good Performance
886	12CC685	66	Low risk	Good Performance
887	12CC686	54	High risk	Average Performance
888	12CC687	56	High risk	Average Performance
889	12CC688	78	Risk free	Excellent Performance
890	12CC689	60	Low risk	Good Performance

891	12CC690	46	Very high risk	Poor Performance
892	12CC691	58	High risk	Average Performance
893	12CC692	74	Risk free	Excellent Performance
894	12CC693	52	High risk	Average Performance
895	12CC694	58	High risk	Average Performance
896	12CC695	60	Low risk	Good Performance
897	12CC696	86	Risk free	Excellent Performance
898	12CC697	70	Risk free	Excellent Performance
899	12CC698	62	Low risk	Good Performance
900	12CC699	68	Low risk	Good Performance
901	12CC700	72	Risk free	Excellent Performance
902	12CC701	66	Low risk	Good Performance
903	12CC702	72	Risk free	Excellent Performance
904	12CC703	64	Low risk	Good Performance
905	12CC704	68	Low risk	Good Performance
906	12CC705	60	Low risk	Good Performance
907	12CC706	64	Low risk	Good Performance
908	12CC707	52	High risk	Average Performance
909	12CC708	58	High risk	Average Performance
910	12CC709	60	Low risk	Good Performance
911	12CC710	68	Low risk	Good Performance
912	12CC711	74	Risk free	Excellent Performance
913	12CC712	54	High risk	Average Performance
914	12CC713	66	Low risk	Good Performance
915	12CC714	42	Very high risk	Poor Performance
916	12CC715	56	High risk	Average Performance
917	12CC716	66	Low risk	Good Performance
918	12CC717	56	High risk	Average Performance
919	12CC718	56	High risk	Average Performance
920	12CC719	42	Very high risk	Poor Performance
921	12CC720	46	Very high risk	Poor Performance
922	12CC721	48	Very high risk	Poor Performance
923	12CC722	48	Very high risk	Poor Performance
924	12CC723	52	High risk	Average Performance
925	12CC724	72	Risk free	Excellent Performance
926	12CC725	46	Very high risk	Poor Performance
927	12CC726	40	Very high risk	Poor Performance
928	12CC727	38	Very high risk	Poor Performance
929	12CC728	60	Low risk	Good Performance
930	12CC729	60	Low risk	Good Performance
931	12CC730	60	Low risk	Good Performance
932	12CC731	58	High risk	Average Performance
933	12CC732	42	Very high risk	Poor Performance
934	12CC733	46	Very high risk	Poor Performance
935	12CC734	38	Very high risk	Poor Performance

936	12CC735	42	Very high risk	Poor Performance
937	12CC736	46	Very high risk	Poor Performance
938	12CC737	68	Low risk	Good Performance
939	12CC738	44	Very high risk	Poor Performance
940	12CC739	46	Very high risk	Poor Performance
941	12CC740	40	Very high risk	Poor Performance
942	12CC741	44	Very high risk	Poor Performance
943	12CC742	52	High risk	Average Performance
944	12CC743	62	Low risk	Good Performance
945	12CC744	82	Risk free	Excellent Performance
946	12CC745	72	Risk free	Excellent Performance
947	12CC746	44	Very high risk	Poor Performance
948	12CC747	72	Risk free	Excellent Performance
949	12CC748	60	Low risk	Good Performance
950	12CC749	52	High risk	Average Performance
951	12CC750	74	Risk free	Excellent Performance
952	12CC751	50	High risk	Average Performance
953	12CC752	84	Risk free	Excellent Performance
954	12CC753	52	High risk	Average Performance
955	12CC754	56	High risk	Average Performance
956	12CC755	56	High risk	Average Performance
957	12CC756	62	Low risk	Good Performance
958	12CC757	82	Risk free	Excellent Performance
959	12CC758	60	Low risk	Good Performance
960	12CC759	56	High risk	Average Performance
961	12CC760	84	Risk free	Excellent Performance
962	12CC761	72	Risk free	Excellent Performance
963	12CC762	52	High risk	Average Performance
964	12CC763	68	Low risk	Good Performance
965	12CC764	64	Low risk	Good Performance
966	12CC765	44	Very high risk	Poor Performance
967	12CC766	72	Risk free	Excellent Performance
968	12CC767	82	Risk free	Excellent Performance
969	12CC768	66	Low risk	Good Performance
970	12CC769	62	Low risk	Good Performance
971	12CC770	42	Very high risk	Poor Performance
972	12CC771	74	Risk free	Excellent Performance
973	12CC772	62	Low risk	Good Performance
974	12CC773	48	Very high risk	Poor Performance
975	12CC774	70	Risk free	Excellent Performance
976	12CC775	64	Low risk	Good Performance
977	12CC776	46	Very high risk	Poor Performance
978	12CC777	52	High risk	Average Performance
979	12CC778	64	Low risk	Good Performance
980	12CC779	84	Risk free	Excellent Performance

981	12CC780	44	Very high risk	Poor Performance
982	12CC781	54	High risk	Average Performance
983	12CC782	78	Risk free	Excellent Performance
984	12CC783	66	Low risk	Good Performance
985	12CC784	54	High risk	Average Performance
986	12CC785	50	High risk	Average Performance
987	12CC786	50	High risk	Average Performance
988	12CC787	62	Low risk	Good Performance
989	12CC788	60	Low risk	Good Performance
990	12CC789	68	Low risk	Good Performance
991	12CC790	48	Very high risk	Poor Performance
992	12CC791	52	High risk	Average Performance
993	12CC792	50	High risk	Average Performance
994	12CC793	64	Low risk	Good Performance
995	12CC794	48	Very high risk	Poor Performance
996	12CC795	50	High risk	Average Performance
997	12CC796	46	Very high risk	Poor Performance
998	12CC797	54	High risk	Average Performance
999	12CC798	42	Very high risk	Poor Performance
1000	12CC799	60	Low risk	Good Performance
1001	12CC800	74	Risk free	Excellent Performance
1002	12CC801	48	Very high risk	Poor Performance
1003	12CC802	78	Risk free	Excellent Performance
1004	12CC803	40	Very high risk	Poor Performance
1005	12CC804	56	High risk	Average Performance
1006	12CC805	50	High risk	Average Performance
1007	12CC806	56	High risk	Average Performance
1008	12CC807	52	High risk	Average Performance
1009	12CC808	66	Low risk	Good Performance
1010	12CC809	46	Very high risk	Poor Performance
1011	12CC810	44	Very high risk	Poor Performance
1012	12CC811	42	Very high risk	Poor Performance
1013	12CC812	52	High risk	Average Performance
1014	12CC813	32	Very high risk	Poor Performance
1015	12CC814	56	High risk	Average Performance
1016	12CC815	38	Very high risk	Poor Performance
1017	12CC816	48	Very high risk	Poor Performance
1018	12CC817	48	Very high risk	Poor Performance
1019	12CC818	46	Very high risk	Poor Performance
1020	12CC819	42	Very high risk	Poor Performance
1021	12CC820	38	Very high risk	Poor Performance
1022	12CC821	54	High risk	Average Performance
1023	12CC822	68	Low risk	Good Performance
1024	12CC823	50	High risk	Average Performance
1025	12CC824	56	High risk	Average Performance

1026	12CC825	42	Very high risk	Poor Performance
1027	12CC826	58	High risk	Average Performance
1028	12CC827	46	Very high risk	Poor Performance
1029	12CC828	62	Low risk	Good Performance
1030	12CC829	64	Low risk	Good Performance
1031	12CC830	40	Very high risk	Poor Performance
1032	12CC831	52	High risk	Average Performance
1033	12CC832	64	Low risk	Good Performance
1034	12CC833	46	Very high risk	Poor Performance
1035	12CC834	42	Very high risk	Poor Performance
1036	12CC835	48	Very high risk	Poor Performance
1037	12CC836	36	Very high risk	Poor Performance
1038	12CC837	46	Very high risk	Poor Performance
1039	12CC838	64	Low risk	Good Performance
1040	12CC839	50	High risk	Average Performance
1041	12CC840	48	Very high risk	Poor Performance
1042	12CC841	46	Very high risk	Poor Performance
1043	12CC842	54	High risk	Average Performance
1044	12CC843	38	Very high risk	Poor Performance
1045	12CC844	44	Very high risk	Poor Performance
1046	12CC845	38	Very high risk	Poor Performance
1047	12CC846	40	Very high risk	Poor Performance
1048	12CC847	50	High risk	Average Performance
1049	12CC848	64	Low risk	Good Performance
1050	12CC849	52	High risk	Average Performance
1051	12CC850	50	High risk	Average Performance
1052	12CC851	50	High risk	Average Performance
1053	12CC852	36	Very high risk	Poor Performance
1054	12CC853	40	Very high risk	Poor Performance
1055	12CC854	40	Very high risk	Poor Performance
1056	12CC855	46	Very high risk	Poor Performance
1057	12CC856	46	Very high risk	Poor Performance
1058	12CC857	32	Very high risk	Poor Performance
1059	12CC858	42	Very high risk	Poor Performance
1060	12CC859	38	Very high risk	Poor Performance
1061	12CC860	50	High risk	Average Performance
1062	12CC861	32	Very high risk	Poor Performance
1063	12CC862	58	High risk	Average Performance
1064	12CC863	44	Very high risk	Poor Performance
1065	12CC864	50	High risk	Average Performance
1066	12CC865	46	Very high risk	Poor Performance
1067	12CC866	48	Very high risk	Poor Performance
1068	12CC867	52	High risk	Average Performance
1069	12CC868	56	High risk	Average Performance
1070	12CC869	40	Very high risk	Poor Performance

1071	12CC870	40	Very high risk	Poor Performance
1072	12CC871	52	High risk	Average Performance
1073	12CC872	48	Very high risk	Poor Performance
1074	12CC873	34	Very high risk	Poor Performance
1075	12CC874	42	Very high risk	Poor Performance
1076	12CC875	44	Very high risk	Poor Performance
1077	12CC876	46	Very high risk	Poor Performance
1078	12CC877	58	High risk	Average Performance
1079	12CC878	40	Very high risk	Poor Performance
1080	12CC879	44	Very high risk	Poor Performance
1081	12CC880	46	Very high risk	Poor Performance
1082	12CC881	40	Very high risk	Poor Performance
1083	12CC882	34	Very high risk	Poor Performance
1084	12CC883	36	Very high risk	Poor Performance
1085	12CC884	68	Low risk	Good Performance
1086	12CC885	44	Very high risk	Poor Performance
1087	12CC886	58	High risk	Average Performance
1088	12CC887	46	Very high risk	Poor Performance
1089	12CC888	54	High risk	Average Performance
1090	12CC889	76	Risk free	Excellent Performance
1091	12CC890	50	High risk	Average Performance
1092	12CC891	70	Risk free	Excellent Performance
1093	12CC892	62	Low risk	Good Performance
1094	12CC893	48	Very high risk	Poor Performance
1095	12CC894	72	Risk free	Excellent Performance
1096	12CC895	54	High risk	Average Performance
1097	12CC896	70	Risk free	Excellent Performance
1098	12CC897	56	High risk	Average Performance
1099	12CC898	72	Risk free	Excellent Performance
1100	12CC899	68	Low risk	Good Performance
1101	12CC900	64	Low risk	Good Performance
1102	12CC901	50	High risk	Average Performance
1103	12CC902	52	High risk	Average Performance
1104	12CC903	38	Very high risk	Poor Performance
1105	12CC904	50	High risk	Average Performance
1106	12CC905	62	Low risk	Good Performance
1107	12CC906	34	Very high risk	Poor Performance
1108	12CC907	60	Low risk	Good Performance
1109	12CC908	48	Very high risk	Poor Performance
1110	12CC909	66	Low risk	Good Performance
1111	12CC910	56	High risk	Average Performance
1112	12CC911	54	High risk	Average Performance
1113	12CC912	60	Low risk	Good Performance
1114	12CC913	66	Low risk	Good Performance
1115	12CC914	60	Low risk	Good Performance

1116	12CC915	42	Very high risk	Poor Performance
1117	12CC916	70	Risk free	Excellent Performance
1118	12CC917	52	High risk	Average Performance
1119	12CC918	34	Very high risk	Poor Performance
1120	12CC919	48	Very high risk	Poor Performance
1121	12CC920	86	Risk free	Excellent Performance
1122	12CC921	40	Very high risk	Poor Performance
1123	12CC922	60	Low risk	Good Performance
1124	12CC923	74	Risk free	Excellent Performance
1125	12CC924	60	Low risk	Good Performance
1126	12CC925	70	Risk free	Excellent Performance
1127	12CC926	74	Risk free	Excellent Performance
1128	12CC927	72	Risk free	Excellent Performance
1129	12CC928	76	Risk free	Excellent Performance
1130	12CC929	40	Very high risk	Poor Performance
1131	12CC930	48	Very high risk	Poor Performance
1132	12CC931	68	Low risk	Good Performance
1133	12CC932	64	Low risk	Good Performance
1134	12CC933	48	Very high risk	Poor Performance
1135	12CC934	58	High risk	Average Performance
1136	12CC935	54	High risk	Average Performance
1137	12CC936	56	High risk	Average Performance
1138	12CC937	56	High risk	Average Performance
1139	12CC938	62	Low risk	Good Performance
1140	12CC939	76	Risk free	Excellent Performance
1141	12CC940	46	Very high risk	Poor Performance
1142	12CC941	56	High risk	Average Performance
1143	12CC942	44	Very high risk	Poor Performance
1144	12CC943	74	Risk free	Excellent Performance
1145	12CC944	70	Risk free	Excellent Performance
1146	12CC945	46	Very high risk	Poor Performance
1147	12CC946	74	Risk free	Excellent Performance
1148	12CC947	78	Risk free	Excellent Performance
1149	12CC948	66	Low risk	Good Performance
1150	12CC949	88	Risk free	Excellent Performance
1151	12CC950	64	Low risk	Good Performance
1152	12CC951	40	Very high risk	Poor Performance
1153	12CC952	42	Very high risk	Poor Performance
1154	12CC953	42	Very high risk	Poor Performance
1155	12CC954	66	Low risk	Good Performance
1156	12CC955	80	Risk free	Excellent Performance
1157	12CC956	30	Very high risk	Poor Performance
1158	12CC957	82	Risk free	Excellent Performance
1159	12CC958	62	Low risk	Good Performance
1160	12CC959	42	Very high risk	Poor Performance

1161	12CC960	62	Low risk	Good Performance
1162	12CC961	64	Low risk	Good Performance
1163	12CC962	62	Low risk	Good Performance
1164	12CC963	48	Very high risk	Poor Performance
1165	12CC964	58	High risk	Average Performance
1166	12CC965	70	Risk free	Excellent Performance
1167	12CC966	58	High risk	Average Performance
1168	12CC967	68	Low risk	Good Performance
1169	12CC968	80	Risk free	Excellent Performance
1170	12CC969	66	Low risk	Good Performance
1171	12CC970	40	Very high risk	Poor Performance
1172	12CC971	38	Very high risk	Poor Performance
1173	12CC972	48	Very high risk	Poor Performance
1174	12CC973	70	Risk free	Excellent Performance
1175	12CC974	40	Very high risk	Poor Performance
1176	12CC975	62	Low risk	Good Performance
1177	12CC976	50	High risk	Average Performance
1178	12CC977	60	Low risk	Good Performance
1179	12CC978	48	Very high risk	Poor Performance
1180	12CC979	68	Low risk	Good Performance
1181	12CC980	56	High risk	Average Performance
1182	12CC981	72	Risk free	Excellent Performance
1183	12CC982	58	High risk	Average Performance
1184	12CC983	48	Very high risk	Poor Performance
1185	12CC984	64	Low risk	Good Performance
1186	12CC985	66	Low risk	Good Performance
1187	12CC986	64	Low risk	Good Performance
1188	12CC987	48	Very high risk	Poor Performance
1189	12CC988	60	Low risk	Good Performance
1190	12CC989	62	Low risk	Good Performance
1191	12CC990	58	High risk	Average Performance
1192	12CC991	70	Risk free	Excellent Performance
1193	12CC992	46	Very high risk	Poor Performance
1194	12CC993	62	Low risk	Good Performance
1195	12CC994	64	Low risk	Good Performance
1196	12CC995	58	High risk	Average Performance
1197	12CC996	70	Risk free	Excellent Performance
1198	12CC997	52	High risk	Average Performance
1199	12CC998	46	Very high risk	Poor Performance
1200	12CC999	54	High risk	Average Performance