STUDENTS DATA CLASSIFICATION MODEL

By

ILI AIMIE SETU(4637)

Final Report
Submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
Business Information Systems

November 20th, 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

STUDENTS DATA CLASSIFICATION MODEL

By

ILI AIMIE SETU

Final Report
Submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
Business Information Systems

November 20th, 2006

Approved by,

____

Ms. Vivian Yong Suet Peng
Academic Supervisor

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledges, and that the original work contained herein have not been undertaken or done by unspecified resources or persons.

Ili Aimie Bt. Setu

ID: 4637

ABSTRACT

In this project, research is conducted based on data sets of undergraduates at varsity level to classify student performance data. The objective of the project is to develop a system that utilizes various intelligent techniques with targeted accuracy being at a minimal level of 88%.

The system is designed to predict students' CGPA upon graduation. Any further actions that can be taken to avoid students' dismissals, or to strengthen their area of interest or expertise can be derived from the outcome of this intelligent system.

The project is implemented using data sets Iris and Student. Techniques used to support classification are separated into two different subprojects: (1) Back propagation feed forward neural network using Bayes probability to initialize weights, and (2) Fuzzy system. The proposed optimization of neural network and Bayes Theorem returns 92.55% level of accuracy for the student data. Further improvements can be performed on areas such as the individual variations of each technique and the combination of all three techniques to optimize accuracy.

The project contributes in customizing a grading system for Universiti Teknologi PETRONAS. This system structure is generally relevant to many universities in Malaysia as they adopt a fairly similar approach in grading.

ACKNOWLEDGEMENT

First and foremost, I'd like to express my greatest gratitude to my project supervisor, Ms. Vivian Yong for her continuous support throughout the duration of this project. Without her guidance and willingness to endure the learning process with me, I would not have been able to complete the project. To Ms. Yong, thank you very much.

Not to forget, Universiti Teknologi PETRONAS(UTP) for the chance to expand my knowledge; for the on-campus facilities, for the wonderful helping hands by several parties who are not directly involved in the project. To the Academic Central Services(ACS); Puan Suhaidah Ismail and Puan Yusmiza Abd. Jabar; for their contribution in this project.

Last but not least, to my parents, Encik Setu Sikin, and Puan Noorlee Tasruhin; for their unrelenting faith in me. For your understanding and unconditional support; for always being there; I thank you very much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

WSBA	Weighted subsethood-based algorithm
QSBA	Quantifier subsethood-based algorithm

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

1.1 Background of Study

Today, most organizations are overwhelmed with a large amount of valuable data that requires proper management and handling in order to reveal sets of undiscovered knowledge. A set of data that is not transformed is considered a huge waste as the execution of specific processes will result in valuable knowledge that can be used to determine a firm's direction in making business decisions. To utilize this meaningless data, it needs to undergo several transformation and mining processes.

Data Mining can be defined as the nontrivial of implicit, previously unknown, and potentially useful information from data[1]. The term "Data Mining" is interchangeably used with the term "Knowledge Discovery in Database" (KDD) and is widely accepted due to the concept fundamentally similar concepts behind both terms. Closely related to Data Mining are the techniques that are used to support Data Mining activities, and examples of these include Artificial Neural Network(ANN), Fuzzy Logic, Genetic Algorithm(GA) and a myriad of clustering techniques. In this project, the aim is to conduct a research on the impact of the combination of fuzzy logic, neural network, as well as Bayes theory on the outcome's accuracy level.

1.2 Problem Statement

Data Mining approach is widely used by business firms from different sectors around the globe to maximize profits hence creating high firm value. This usually focuses on customer retention campaigns and customized marketing strategies according to customer profiles respectively. However, looking at the same concept from the pedagogical perspective, Data Mining can also be applied to track students' performances. In this case, the research is to be conducted based on data sets of undergraduates at varsity level. Specifically, Extract, Transform, Load(ETL) method is applied on data sets of a portion of undergraduates to produce meaningful knowledge as the final product. Some of the commonly faced problems by academicians and students alike are as further discussed in the following sections.

Amongst the relevant issues are not being able to identify each student's areas of strengths and weaknesses in a structured manner. This knowledge is clearly beneficial to both students and academicians alike. Through the application of Data Mining methods, students' future performances can be forecasted using prediction. Other than attending to future issues, the system is also designed to cater for current situations where student-academician relationship can be optimized through the knowledge that the system provides. Academicians can learn, improvise and customize his practice by viewing this knowledge and recognize trends. This allows the academicians to formulate the best approach in tutoring. Currently, academicians have no means of designing the best approach in reaching for students. There exists no knowledge baseline that houses student academic history during his course of study in the university. Thus, academicians solely rely on restricted information that is gathered through self-observation, intuition and limited reference to a set of manually-recorded past academic records. This identification of strengths and weaknesses is universal as it can generally be applied to students from all groupings according to students' Cumulative Grade Point Average(CGPA) respectively.

The second issue is more focused on a specific group of students. This particular group has the common trait of students who are to be classified as excellent performers. Currently, there exists no structured system that is attending to this group of students to aid in maintaining their excellent performance if not expanding the ability. In this case, the proposed system serves the purpose of assisting students in maximizing their specific or general overall skills that are relevant to a particular programme. The choice depends on the particular student's prospect and interest, provided that the process is based on academicians' advises and is conducted under strict supervision.

Lastly, the issue is focused on the students who require urgent attention for academic rehabilitation. There is a high probability that these students fall under the academic probationary group where either their Grade Point Average(GPA) or Cumulative GPA(CGPA) is not at par with the university's minimum requirements. In this case, the analysis is narrowly focused on determining the best solution in improving these students' performances. Clearly, this is vital as to avoid dismissals in the future. Other than identifying the weaknesses of these students, another aiding tool includes the change of programme guide where the system is capable of suggesting the most suitable programme for a particular student if he decides to pursue a different major instead of continuing on with the same major. Currently, the rehabilitation session only involves heuristics-based decisions in which are mutually agreed by both students and academicians. It is hoped that the new system is able to provide more accurate decisions as the analysis conducted is based on historical data instead of 88% on human judgment while the remaining is based on incomplete previous academic records.

1.3 Objective and Scope of Study

The objectives of this project are as specified below:

- To develop a system that utilizes fuzzy logic, neural network and Bayes techniques in classification and prediction.
- The utilization of these three techniques to return a result set of minimally 88% accuracy level.

These three techniques are used to illustrate Data Mining concepts through producing accurate knowledge on student performance. As for the scope of study, a sample train data of Universiti Teknologi PETRONAS's undergraduates' results is applied.

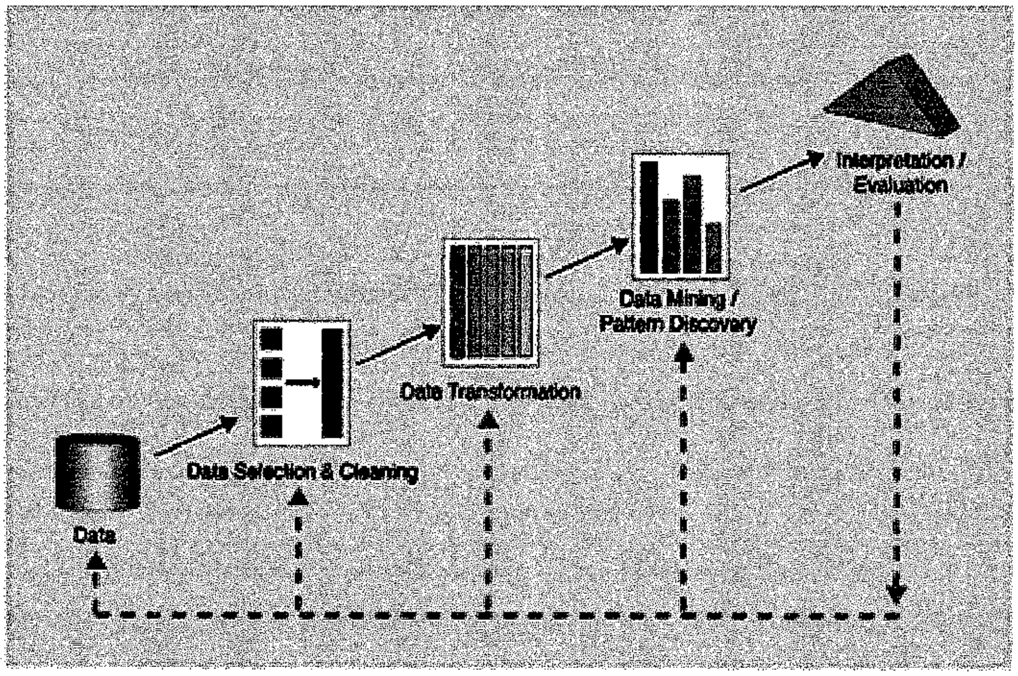
CHAPTER 2

LITERATURE REVIEW AND THEORY

2 LITERATURE REVIEW AND THEORY

2.1 Data Mining

Data Mining or Knowledge Discovery in Databases(KDD) refers to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [2]. Provided below is the diagram of the model used to complete knowledge discovery process.



Source: http://alg.ncsa.uiuc.edu/tools/d2k/manual/dataMining.html

FIGURE 2.1: Knowledge Discovery in Databases(KDD) Process Diagram

The knowledge discovery goals are defined by the intended use of the system. There are two types of goals that can be distinguished: (1) verification and (2) discovery. With verification, the system is limited to verifying the user's hypothesis. With discovery, the system autonomously finds new patterns. Discovery goal is further subdivided into prediction and description [3]. Prediction refers to scenarios when the system finds patterns for forecasting the future behavior of some entities. On the other hand, description refers to scenarios when the system finds patterns for presentation to a user in a human-understandable form. In this particular project, both prediction and description KDD are applied in order to perform classification as well as prediction. The stages as shown in Figure 2.1 are briefly described below:

- *Data Selection and Cleaning*

  Relevant data is selected from data warehouse or legacy system, cleansed in terms of formats, merging, removal of trailing comments etc. prior to transformation process.

- *Data Transformation*

  The Extract, Transform, Load(ETL) procedures are performed at this stage where desired data in specific formats and fulfilling the requirement is gathered in a database known as the Data Mart.

- *Data Mining / Pattern Discovery*

  Analytical processes to classify or predict data are conducted to discover new knowledge.

- *Interpretation and Evaluation*

  Knowledge obtained from Data Mining is manipulated and optimized according to interests accordingly i.e. retain customer, predict student failures etc.

2.2 Fuzzy Logics(FL)

The primary method of assessment usually involves awarding numerical marks by an evaluator. Such marks are usually given according to a given marking scheme. These marks are usually numerical values that may fluctuate a little as different evaluators may award different marks. Evaluation of a student's work may be affected by the evaluator's experience, sensitivity and the standard used. Thus marks awarded by an evaluator to represent student performance are only an approximation. Although linguistic terms (e.g. bad, good, very good, excellent etc.) have also been widely used to represent the final student's performance, their inherent nature of vagueness is often ignored. However, academic performance evaluation involves the measurement of ability, competence and skills. Ability, competence and skills are fuzzy concepts and can be approximately captured in fuzzy terms.[8]

A rule-based system utilizes a model that represents human knowledge in the form of "IF-THEN" rules. This conventional approach has been adapted to build fuzzy rule-based systems. A simple fuzzy IF-THEN rule can be written in the form of " IF x is A THEN y is B" where A and B are fuzzy sets. This can be extended to more than two fuzzy sets resulting in compound fuzzy propositions. In general, fuzzy IF-THEN rules are production rules whose antecedents, consequences or both are fuzzy [12]. Mendel [13] classified fuzzy rules into 6 different types, namely Incomplete Rules, Mixed Rules, Fuzzy Statement Rules, Comparative Rules, Unless Rules and Quantifier Rules. However there is no agreed classification of fuzzy rule models [12] and a single rule might involve a combination of several different classification types [13].

One method of Fuzzy Logics include the IF-THEN method(Mamdani-type). This model also has high interpretability. The structure is as shown below:

IF X_{1i} is A_1 and ... and X_n is A_n **THEN** Y_1 is B_1 and ... and Y_m is B_m

,where X_{1i} are fuzzy input linguistic variables, Y_j are fuzzy output linguistic variables, and A_i and B_j are linguistic terms in the form of fuzzy sets that characterize X_i and Y_j. In order to make a system that is more readily comprehensible to the user, weighted subsethood-based algorithm(WSBA) employs a rule generation algorithm which is based on fuzzy general rules or the extension of a Mamdani-type [8].

Consider fuzzy rules with multi-inputs and a single output. These rules can be written in the following form:

IF A is (A1 *OR* A2 *OR* ...*OR* A_i) **AND** B is (B1 *OR* B2 *OR*... *OR* B_j) **AND** ... **AND** H is (H1 *OR* H2 *OR* ... *OR* H_k) **THEN** the classification output is (X1 *OR* X2 *OR*...*OR* X_n)

This general rule can be re-written in a more specific form with each rule corresponding to one classification output value:

Rule 1

IF A is (A1 *OR* A2 *OR* ...*OR* A_i) **AND** B is (B1 *OR* B2 *OR*... *OR* B_j) **AND** ... **AND** H is (H1 *OR* H2 *OR* ... *OR* H_k) **THEN** the classification output is X1

Rule 2

IF A is (A1 *OR* A2 *OR* ...*OR* A_i) **AND** B is (B1 *OR* B2 *OR*... *OR* B_j) **AND** ... **AND** H is (H1 *OR* H2 *OR* ... *OR* H_k) **THEN** the classification output is X2

Rule n

IF A is (A1 *OR* A2 *OR* ...*OR* A_i) **AND** B is (B1 *OR* B2 *OR*... *OR* B_j) **AND** ... **AND** H is (H1 *OR* H2 *OR* ... *OR* H_k) **THEN** the classification output is X_n

Thus, all linguistic terms of each attribute are used to describe the antecedent of each rule initially. This may look tedious, but the reason for keeping this complete form is that every linguistic term may contain important information that should be taken into account. Otherwise, there is no need for adopting the given fuzzy partitions of the underlying domains in the first place. Of course, during training, some of such terms may be omitted due to no evaluated contribution (or with a relative weight of 0) with regard to the training data.

However, the above default rules do not tell any differences between the relative contributions made by the individual linguistic terms of each variable towards the eventual conclusion drawn. It is here that relative weights computed via subsethood values can help. Following this idea, by multiplying each linguistic term by its respective weight, the fuzzy rules to be generated will be of the form:

Rule 1

IF A is (w(X1,A1)A1 *OR* (w(X1,A2)A2 *OR* ...*OR* w(X1,A_i)A_i) **AND** B is (w(X1,B1)B1 *OR* w(X1,B2)B2 *OR*... *OR* w(X1,B_j)B_j) **AND** ... **AND** H is (w(X1,H1)H1 *OR* w(X1,H2)H2 *OR* ... *OR* w(X1,H_k)Hk) **THEN** the classification output is X1

Rule 2

IF A is (w(X2,A1)A1 *OR* (w(X2,A2)A2 *OR* ...*OR* w(X2,A_i)Ai) **AND** B is (w(X2,B1)B1 *OR* w(X2,B2)B2 *OR*... *OR* w(X2,B_j)B_j) **AND** ... **AND** H is (w(X2,H1)H1 *OR* w(X2,H2)H2 *OR* ... *OR* w(X2,H_k)H_k) **THEN** the classification output is X2

...

Rule n

IF A is (w(X_n,A1)A1 *OR* (w(X_n,A2)A2 *OR* ...*OR* w(X_n,A_i)A_i) **AND** B is (w(X_n,B1)B1 *OR*
w(X_n,B2)B2 *OR*... *OR* w(X_n,B_j)B_j) **AND** ... **AND** H is (w(X_n,H1)H1 *OR* w(X_n,H2)H2 *OR* ... *OR* w(X_n,H_k)H_k) **THEN** the classification output is X_n

The weights for each linguistic term are considered as a quantifier "some" or "all". If the weight = 1, the quantifier is regarded to be "all", otherwise it is considered to represent "some". The extent to which "some" is interpreted depends on the value of the weights of the respective linguistic terms. Researchers suggested many different types of hedges for fuzzy systems. Among the discussed methods are concentration and dilution[23]. Figure 2.2 illustrates the concentration concept. The normal membership segment is shrunk to focus on a smaller segment of the fuzzy set.

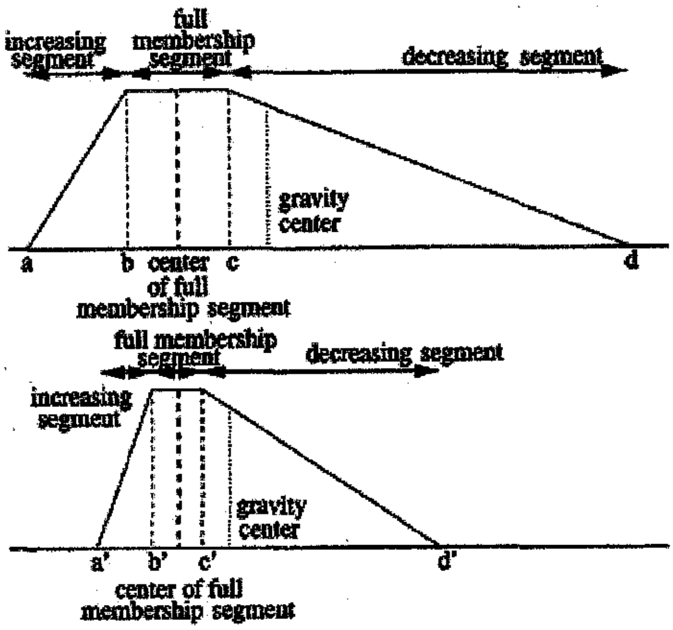


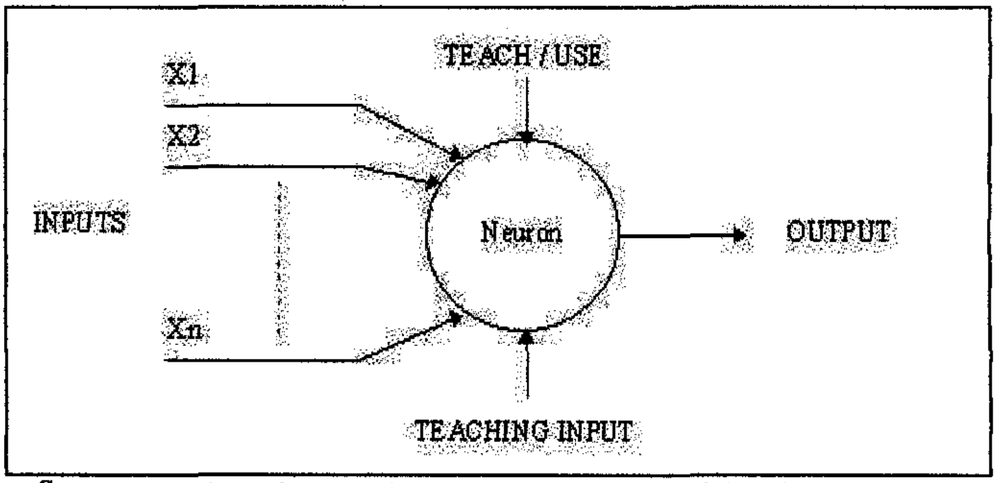
FIGURE 2.2: Sample of concentration hedge

2.3 Artificial Neural Network(ANN)

Generally, an artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not[7]. Figure 2.3 illustrates the structure of a simple neuron.

Information is stored in the weight matrix W of a neural network. Learning is the determination of the weights. Following the way learning is performed, two major categories of neural networks can be distinguished:

- fixed networks in which the weights cannot be changed, ie dW/dt=0. In such networks, the weights are fixed according to the problem to solve.
- adaptive networks which are able to change their weights, ie dW/dt not= 0.



Source: *http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html*

FIGURE 2.3: Simple neuron structure

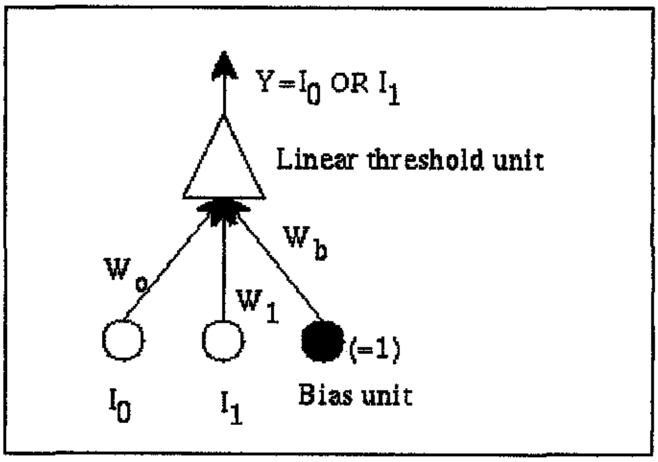
As can be seen from Figure 2.4, the network has 3 inputs, and one output. All numbers are in binary format. The node is to learn simple OR discrete mathematics concept: output is 1 if either I_0 or I_1 is 1. The output is

if $W_0 * I_0 + W_1 * I_1 + W_b > 0$ then

0

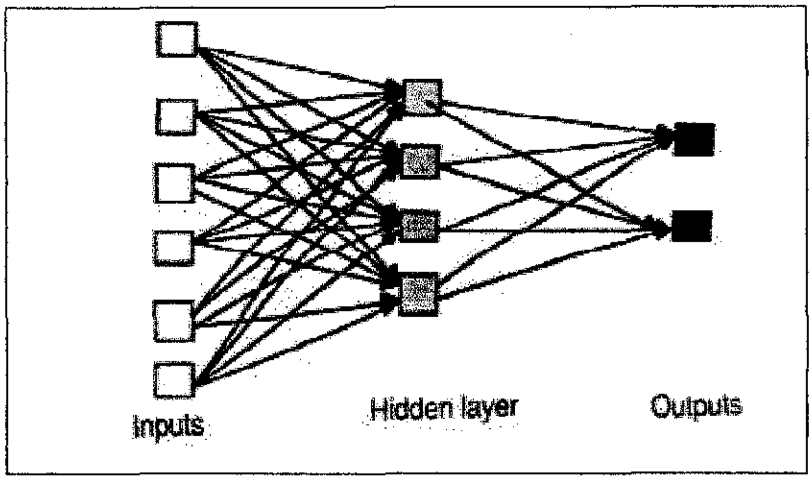
if $W_0 * I_0 + W_1 * I_1 + W_b <= 0$ then

1 (2.1)



Source: http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html

FIGURE 2.4: Adaptive network simple unit.



Source: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

FIGURE 2.5: Feed forward ANN

There are two types of ANN architectures. Firstly, feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down. Secondly, feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations. [7]

One of the most popular neural network architectures used for classification is the Multi-Layer Perceptron. The units are organized into different layers, and the network is said to be feed-forward because the activation values propagate in one direction only, from the units in the input layer, through a number of hidden layers, to end up in the output layer. The multi-layer perceptron is usually trained with the Error Back-Propagation method. Initially the weights in the network are set randomly. The training samples are fed one at a time into the input layer and the activity propagated through the network to the output layer. The output of the network is then compared to the desired, and the difference gives rise to an error signal which is fed backwards through the network, causing the weights to be updated in a way which will decrease the error the next time the same pattern is presented. By going through the training set in this way several times, the weights are gradually adjusted to minimize the output error[17]. Generally, the complete rule for modifying the weight W_{AB} between a neuron A sending a signal to a neuron B is,

$$W_{AB(new)} = W_{AB(old)} - \eta \frac{\partial E^2}{\partial I_B} O_A \tag{2.2}$$

where,

$$\frac{\partial E^2}{\partial I_B} = 2E\, f_o'(I_B) \qquad \text{-}I_B \text{ is the output neuron}$$

$$\frac{\partial E^2}{\partial I_B} = \frac{\partial E^2}{\partial I_O} W_{BO}\, f_h'(I_B) \qquad \text{-}I_B \text{ is the hidden neuron} \tag{2.3}$$

where f_o and f_h are the output and hidden activation functions respectively.

As suggested by the name, One-Layer Perceptron is the precedence of multi-layer perceptron. The single layer of weights between input and output units is trained, just as in the multi-layer case, with a gradient descent method, which adjusts the weights a small step in the direction which will make the classification of the current pattern more correct.

2.4 Statistical Method: Bayesian Theory.

2.4.1 Previous Findings

Bayes' Theorem, developed by the Rev. Thomas Bayes, an 18th century mathematician and theologian, was first published in 1763 [14]. Here, the probability is calculated where a Bayesian inference can be made based on available information. Mathematically it is expressed as:

$$P(H|E, c) = \frac{P(H|c) * P(E|H,c)}{P(E|c)} \tag{2.4}$$

The left-hand term, P(H|E,c) is known as the "posterior probability," or the probability of H after considering the effect of E on c. The term P(H|c) is called the "prior probability of H given c alone. The term P(E|H,c) is called the "likelihood" and gives the probability of the evidence assuming the hypothesis H and the background information c is true. Finally, the last term P(E|c) is independent of H and can be regarded as a normalizing or scaling factor. It is important to note that all of these probabilities are conditional. They specify the degree of belief in some proposition or propositions based on the assumption that some other propositions are true. As such, the theory has no meaning without prior resolution of the probability of these antecedent propositions[14].

There are two major classification using different Bayesian rules [17]:

- One-layer neural network: deduced Bayesian learning rule.
- Multi-layer neural network: complex columns in hidden layer.
    - Partitioning
    - Overlapping

The assumption underlying the Naive Bayesian Classifier is that all input attributes are independent. Then the probability distribution over the domain can be written as a product of the marginal distributions over the attributes. These marginal distributions have much fewer parameters, and are thus much easier to estimate from the training data. The independence assumption amounts to assuming that each input attribute gives some evidence for or against each class, which can be considered separately from the evidence contributed by the other attributes. The one-layer Bayesian neural network is based on the idea of a naive Bayesian classifier. The network is trained according to the Bayesian learning rule, which considers the units in the network as representing stochastic events, and calculates the weights based on the correlation between these events. The activity of a unit is interpreted as the probability of that event, given the events corresponding to already activated units. The equation is as displayed in Eq. 2.2.

When all input attributes are not independent, the naive Bayesian classifier is not appropriate. If it were possible it is better to estimate the whole distribution $P(X|Y)$ directly, but this is impossible already for moderate numbers of input attributes. The solution that is concentrated on is to make something in between. Those attributes that are dependent must be considered together, but hopefully every attribute is not dependent on all others. There are two different ways to handle dependencies. The first method is to partition the input attributes into groups which are independent. Each group can be considered as one complex attribute, and the joint distribution over it is estimated. Since the different groups are independent of each other, their probability distributions can then be combined as before. The second method is more involved. It consists of trying to estimate a dependency graph between the input attributes, and uses this graph to calculate the joint probability distribution over the whole input space[17].

Researchers have made a substantial amount of effort to improve naive Bayes. Related work can be broadly divided into two approaches: eager learning and lazy learning, depending on when the major computation occurs. Eager learning does major computation at training time. Different from eager learning, lazy learning spends little or no effort during training and delaying computation until classification time[5]. An example is Hill Climbing method.

2.5 Researches on Student Data

Many research papers [26], [8], [24], [25] focuses on fuzzy techniques in mining student data. To improve the basic fuzzy technique, quantifiers, and neural techniques are used as supporting techniques. Rasmani has researched on the latest improvisation technique called fuzzy quantifier subsethood-based rule algorithm(FuzzyQSBA). The fuzzy membership value degrees obtained using fuzzy rule-based approach is used to determine how strong the student performance belongs to a specific letter-grade[24]. Table 2.1 shows the original grade and the new grade as transformed using FuzzyQSBA.

TABLE 2.1: FuzzyQSBA Approach in transforming grades

	Original Grade	Membership Value Degree					New Grade
		E	D	C	B	A	
1	A	0	0	0	0.233	0.85	A
2	A	0	0	0	0.233	0.5	A
3	A	0	0	0	0.233	0.85	A
4	B	0	0	0	0.25	0.1	B
5	B	0	0	0.2	0.25	0	B
6	B	0	0	0.05	0.25	0	B
7	C	0	0	0.44	0.25	0	C
8	C	0	0	0.367	0.111	0	C
9	C	0	0	0.352	0.15	0	C
10	D	0.16	0.3	0.333	0.036	0	C
11	D	0.2	0.4	0.06	0	0	D
12	D	0.16	0.2	0.167	0.011	0	D
13	E	0.733	0	0	0	0	E
14	E	0.333	0	0	0	0	E
15	E	0.6	0.150	0	0	0	E

[16]

Aside from FuzzyQSBA, NEFCLASS neuro-fuzzy classification is also utilized on student academic data. During the learning process, fuzzy rules are generated based on overlapping rectangular clusters created by the grid representing fuzzy sets for the input variables. Optimization process is conducted through back-propagation where the error rates obtained using the generated fuzzy rules are used to modify the parameters of the initial membership functions that represent fuzzy rule antecedents[24].

CHAPTER 3

METHODOLOGY

3 METHODOLOGY

3.1 System Design & Architecture

The general flow of this project follows the general flow of the waterfall development model. It starts of with requirement definition and analysis, system design, and finally the implementation on specific domains as illustrated in Figure 3.1.

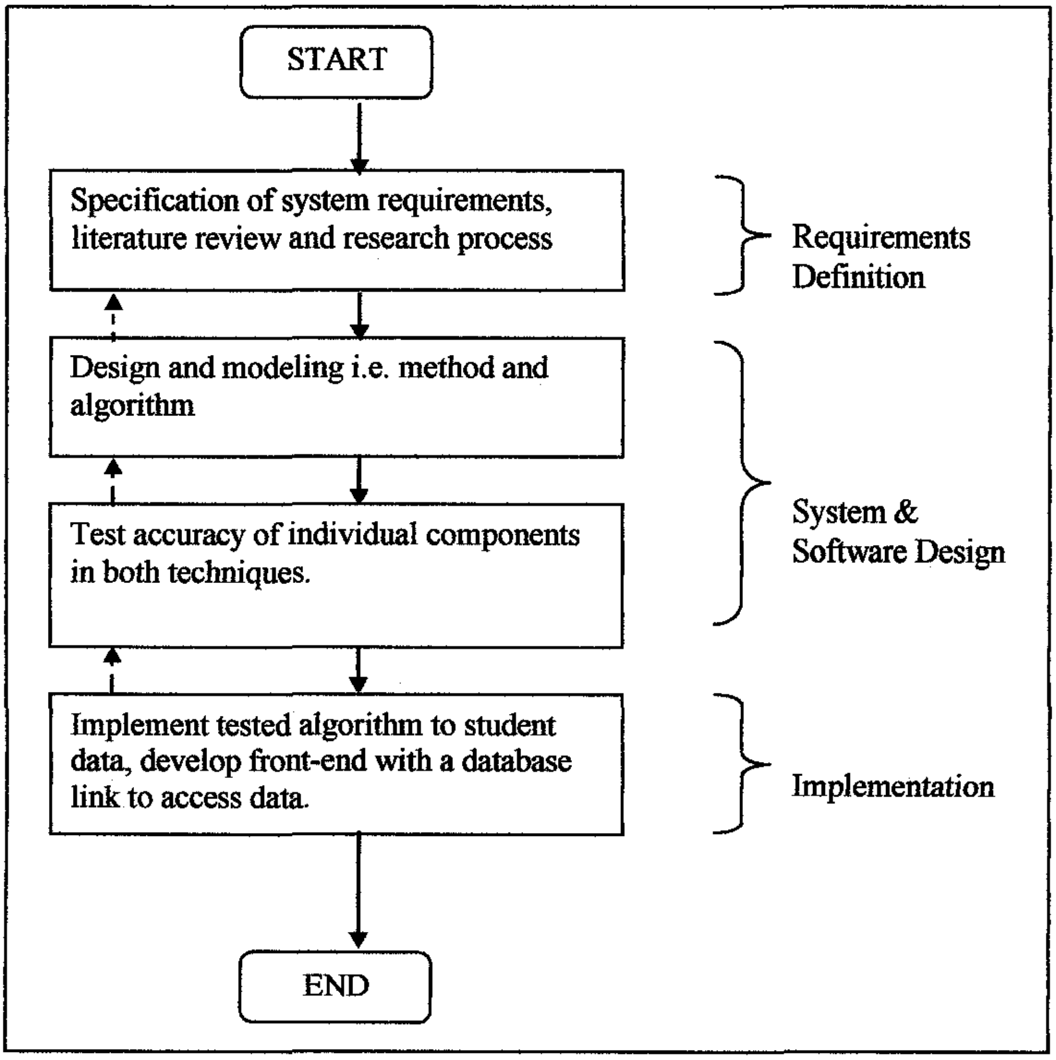


FIGURE 3.1: Flow of Classification System

In this project, two distinctive techniques are used to assist in classification of data; and they are the back propagation feed forward neural network(BPFFNN) and fuzzy logic(FL) techniques. The purpose of this is to observe the accuracy level returned for both techniques and latter implement the most accurate technique on the UTP student data set.

Figure 3.2 shows the architecture in a more comprehensible structure of the system. There are two independent techniques shown in the figure, and these techniques are executed in a parallel manner. Referring to the objective of this project being the utilization of intelligent techniques to return a result set with 88% level of accuracy, testing for both subprojects are equally vital to ensure successful product delivery. However, ensuring that the modeling is accurate is the main priority before venturing into the prototype development phase. This is aligned with the area of research for this project which vastly covers theories and concepts of data mining and intelligent techniques. It is expected that most of the resources of this project is highly focused on the design phase where iteration of testing substages take place within the design phase itself.

3.1.1 System Architecture

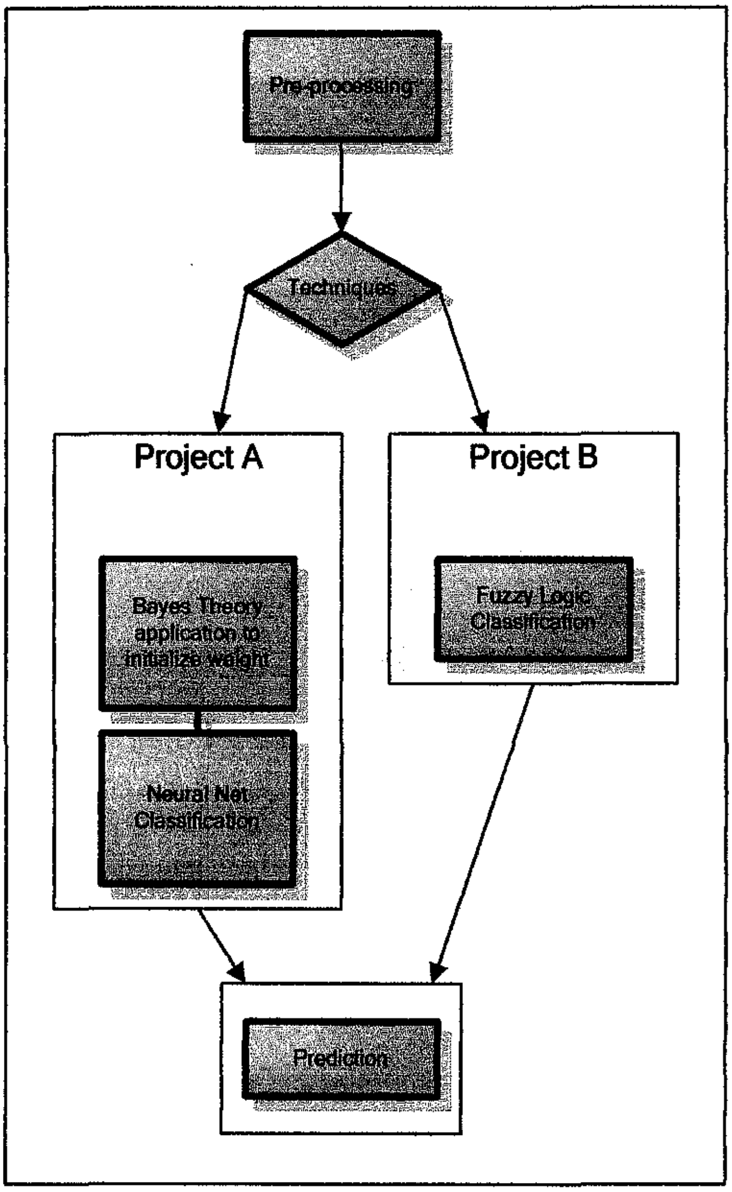


FIGURE 3.2: Architecture of the system

3.1.2 Pre-Processing

This pre-processing procedure only applies for the raw academic data that is obtained from UTP. Publicly available test data sets do not require this process as they are prepared directly for testing purposes. The purpose of pre-processing is to produce data sets of the desired format.

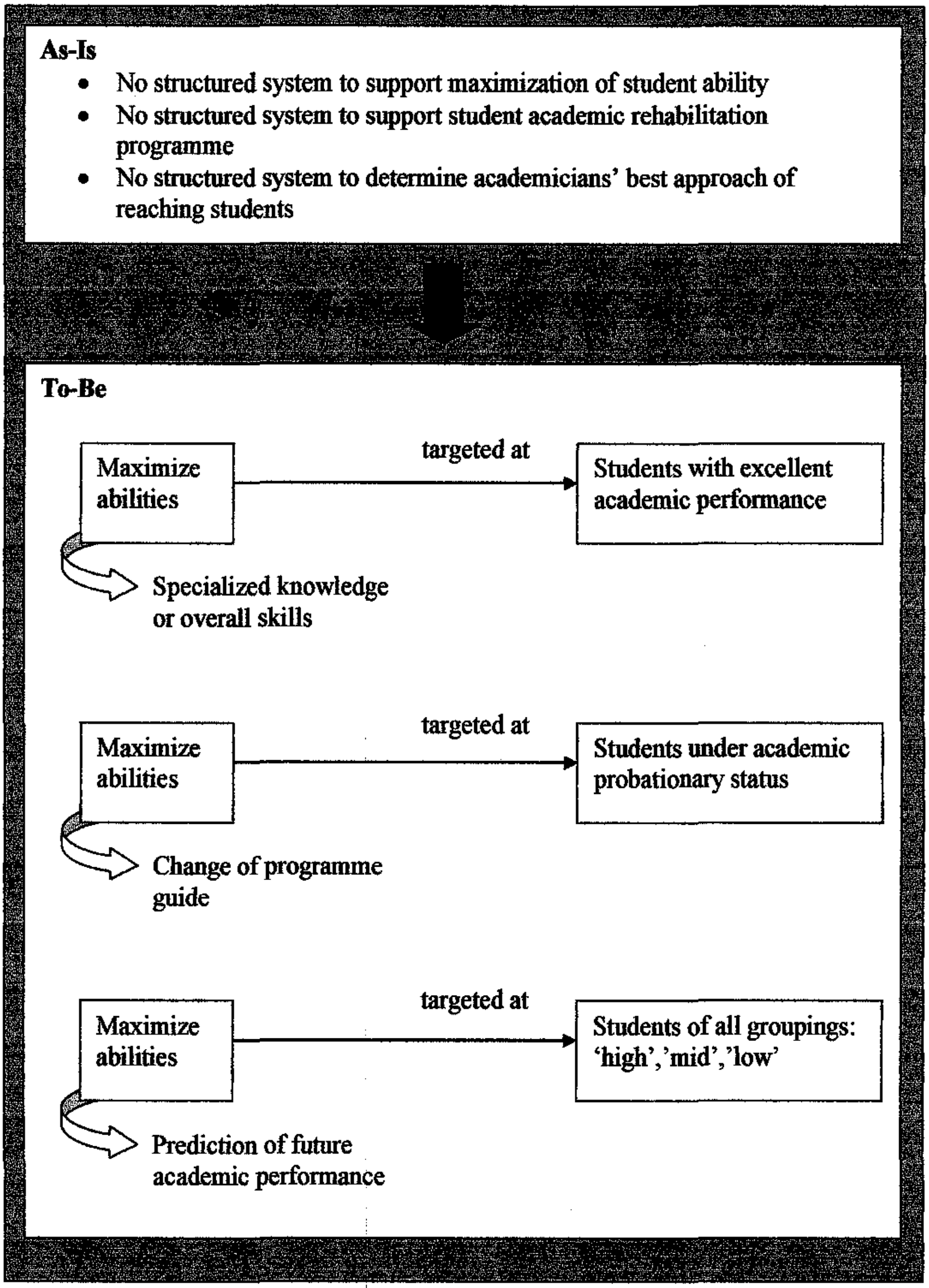


FIGURE 3.3: The relevance of pre-processing stage in the system

UTP Student Data requires extensive patching processes due to factors such as change of structures and special circumstances that leads to outliers. For example, student who changes course during his/her duration of study in UTP may have more subjects in record. Currently, extraction and transformation processes are not fully automated as data cleaning needs massive human intervention in order to allow correct interpretation of data.

Figure 3.3 illustrates how pre-processing takes place in order to create a set of valid and meaningful information. Currently, there is no formal system in UTP that is used to identify the academic status of students. The academic data goes through the process of producing current results in specific values of GPAs and CGPAs. No prediction of mining is currently conducted to perform mining in identifying strengths, or in avoiding dismissals.

The retrieval of raw academic data from ACS is followed by filtering, cleaning and patching up process. Although the number of instances initially provided totals up to 400, the pre-processing stage filters out the invalid and dirty data to produce 300 instances that are ready for testing.

The student raw academic data consists of courses, grades, GPAs, and CGPAs columns. These data is to be grouped into 3 specific fields: (1) Core, (2) Major, (3) Electives. The grades for respective subjects are then summed up and averaged as grouped by these three fields. The two data sets used in this project are Iris and Student. Provided in Table 3.1 are the attributes of these data sets with a specific description on the customized data sets used for this project.

- *Iris Data Set (Source: WEKA Project)*

The Iris data set contains four linguistic variables: sepal length, sepal width, petal length, and petal width. It consists of 150 object instances and 3 output classes: setosa, versicolor, virginica. The considered linguistic terms for each linguistic variable are: small, average, and large.

- *UTP Student Data Set*

The student data set is the domain in which the system is applied on. The source is directly obtained from UTP Academic Central Services. Following pre-processing stage, the data set consists of 300 object instances, and 3 linguistic variables: excellent, good, and poor. The 4 output classes are: first, second upper, second lower, third.

TABLE 3.1: A summary of the used data sets.

Data sets	No of Instances	No of Inputs	No of Outputs	Type of Output	Others
Iris	150	4	3	Continuous	4 measurements in cm, with accuracy of 0.1 cm, petals and sepals of 3 kinds of Iris flowers.
Student	300	3	4	Continuous	Measurements in CGPA mean of 3 subject groups.

To pre-process the raw student data set, basic database operations are conducted. Each subject is tagged with a subject group; being either core, elective or major. For each student, the average grades for all three subject groups are calculated using this query statement:

```
select id, sum(grade)/count(*)
    from sample_data
    group by id, subject_group;
```

An illustration for this process is portrayed in Figure 3.4. In this case, the raw academic data for student tagged as I2010 is collected. To begin with pre-processing, each subject is then tagged with a new column subject_group, where the subject groupings are inserted with core, elective, or major. Then, the mean of grades for these subjects are calculated accordingly using the query as stated in Figure 3.4. The outcome of this should be three columns for student I2010; which store the mean of grade as grouped by respective subject groups. Appendix 3-1 and 3-2 displays the student test data set and the raw academic data respectively.

raw academic data

Subject	I2010
Data Structures	3.75
Computer Organization	3.5
Corporate Finance	3.5
Organizational Behavior	3.75
Knowledge Management	4
Geographical Information Systems	4

Sample_data

Subject_group	subject	I2010
Core	Data Structures	3.75
Core	Computer Organization	3.5
Elective	Corporate Finance	3.5
Elective	Organizational Behavior	3.75
Major	Knowledge Management	4
Major	Geographical Information Systems	4
	CGPA	3.69

```
create table acc_subject_group as select id, sum(I2010)/count(*) from
sample_data group by subject_group, id;
```

acc_subject_group

Id	subject_group	Mean
I2010	CORE	3.69
I2010	ELECTIVE	3.69
I2010	MAJOR	4

FIGURE 3.4: An example of pre-processing for one student instance.

3.1.3 Bayes Theory

Formula 2.4 is applied on all instances to get the Bayes probability of each feature. This process of calculation will generate a total of 36 values as can be derived from the matrix constructed on the three mentioned factors. Database tables are used for this purpose, where initially calculations are conducted based on features and classes, and next combining the values into subresults of the nominator and denominator. The final calculation of Bayes probability includes the combination of all subresults as specifically stated in Formula 2.4. There are two procedures and functions that are developed in order to calculate the Bayes probability. The purpose of the procedure is to calculate set probability subvalues, and the purpose of the function is to get the probability subvalues for final calculation. Displayed in Figure 3.5 is an example of deriving the Bayes probability value for excellent performance in core subjects given a first class degree. Illustrated are also the stages of calculation involved as divided into three database table sets respectively.

$$P(\text{Feature Given Class}) = \frac{\overbrace{P\ (\text{Feature})}^{\text{Table Set 1}} * \overbrace{P\ (\text{Class Given Feature})}^{\text{Table Set 2}}}{\underbrace{P\ (\text{Class})}_{\text{Table Set 3}}}$$

$$P(\text{Core is Excellent Given First}) = \frac{P\ (\text{Core is Excellent}) * P\ (\text{First Given Core is Excellent})}{P\ (\text{First})}$$

FIGURE 3.5: How Bayes probability is derived from pre-processed academic data

The Bayes probability values are then mapped to the number of neurons in the neural network hidden layer in order to allow initialization of the network's weights as opposed to randomly assigning weights to the network. The outcome of this part of the system includes the weights that are assigned to each possible combination of linguistic variables and linguistic terms. Table 3.2 illustrates the structure that is used to calculate individual weights for the Feed Forward Neural Network.

TABLE 3.2: Structure of Student Data Set for Bayes Theory calculation.

Variable Input	Linguistic Term	Class			
		Third	Second Lower	Second Upper	First
Core	Poor	W1 * Core is Poor	W10 * Core is Poor	W19 * Core is Poor	W28 * Core is Poor
Core	Average	W2 * Core is Average	W11 * Core is Average	W20 * Core is Average	W29 * Core is Average
Core	Excellent	W3 * Core is Excellent	W12 * Core is Excellent	W21 * Core is Excellent	W30 * Core is Excellent
Major	Poor	W4 * Major is Poor	W13 * Major is Poor	W22 * Major is Poor	W31 * Major is Poor
Major	Average	W5 * Major is Average	W14 * Major is Average	W23 * Major is Average	W32 * Major is Average
Major	Excellent	W6 * Major is Excellent	W15 * Major is Excellent	W24 * Major is Excellent	W33 * Major is Excellent
Elective	Poor	W7 * Elective is Poor	W16 * Elective is Poor	W25 * Elective is Poor	W34 * Elective is Poor
Elective	Average	W8 * Elective is Average	W17 * Elective is Average	W26 * Elective is Average	W35 * Elective is Average
Elective	Excellent	W9 * Elective is Excellent	W18 * Elective is Excellent	W27 * Elective is Excellent	W36 * Elective is Excellent

3.1.4 Neural Network

A two-layer of hidden network is used as to support neural network architecture. The activation function used is Sigmoid based. The Back Propagation Feed Forward Neural Network(BP FFNN) includes 4 number of inputs, 2 number of hidden layers, and 3 number of outputs. Weights derived from Bayes Theory are assigned to each synapse that connects the neurons. These weights are trained until the error is minimized to a certain extent. The parameters settings that are used to execute the neural network are as specified next. Figure 3.6 shows the architecture of the neural network, with 3-4-3-4 combination of neurons in the layers.

- Learning rate is set to 0.8
- Halt training if maximum epochs is equal to 50000 or error ratio is less than 0.05
- Data is divided; training 60%, cross validation 20%, testing 20%
- Activation function is sigmoidal.
- 2 hidden layers, with 3 neurons in the input layer, and 4 neurons in the output layer.
- The folds for data are set as randomized ordering of instances.

- The training is conducted for 20 cycles.

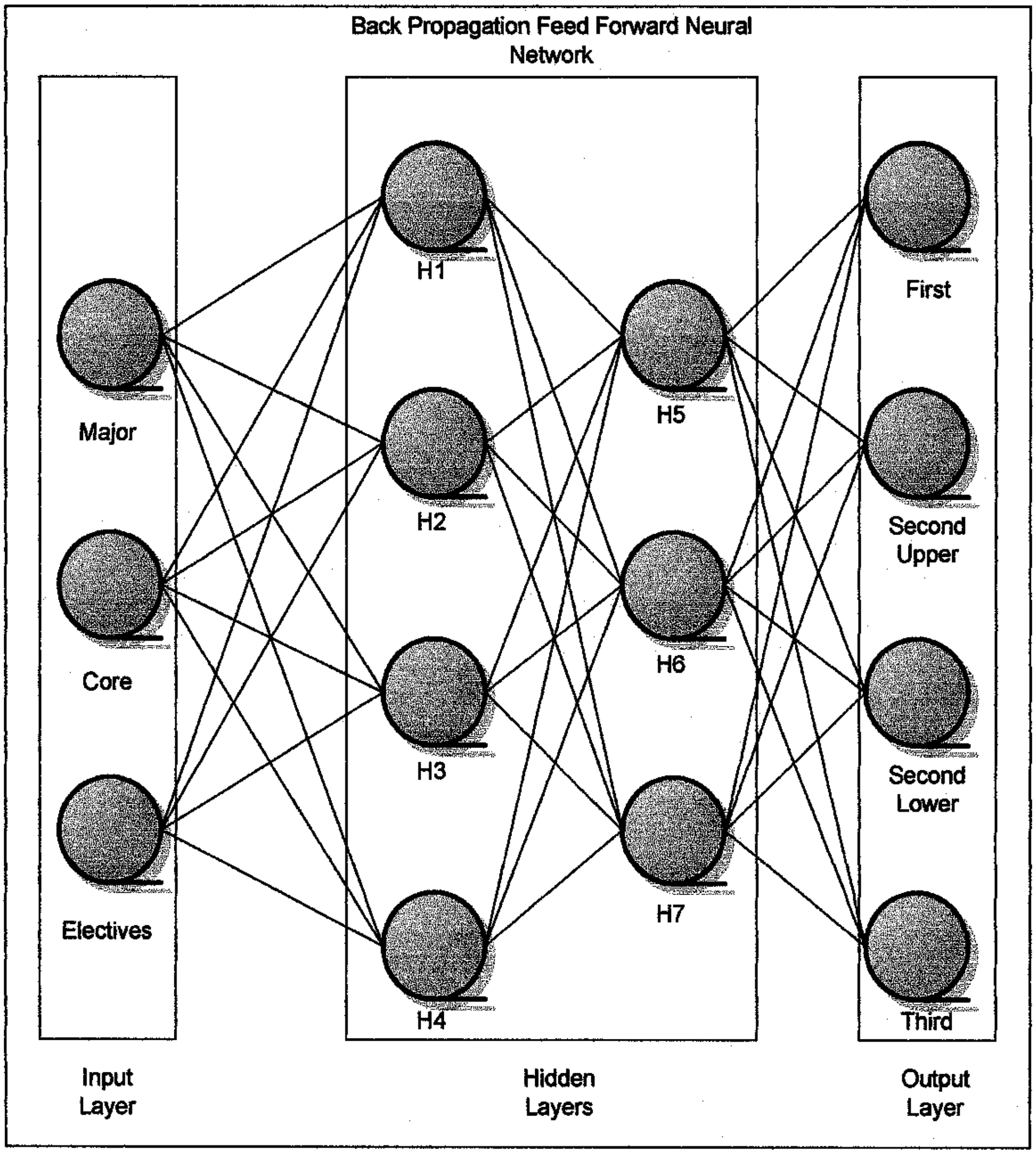


FIGURE 3.6: Architecture of back propagation feed forward neural network

The learning process for the BP FFNN is conducted based on the reassignment of weights as the weights at the output layer is back propagated to the previous layers until the minimized error rate is reached. In this particular project, there are 2 conditions that halt training: (1) When the maximum number of epochs is equal to

50000, or (2) Error ratio is less than 0.05. The formulae for this weight modification can be viewed from formulae 2.2 and 2.3.

3.1.5 Fuzzy Logic

3.1.5.1 Input and Definition of membership function

At this stage, the data set values are fully loaded for further transformation processes. Following the loading, fuzzy variables which include Core, Major and Elective are then declared. As for testing data set Iris, the fuzzy variables are Petal Length, Petal Width, Sepal Length and Sepal Width.

Fuzzy Sets are then defined to identify the degree of membership of each variable. Following this, membership functions are constructed depending on the number of linguistic terms and the number of classes that a data set generates. An example code snippet is provided below where variable Core is declared with three linguistic terms: (1) Excellent, (2) Good, and (3) Poor.

```
Core.addTerm("Excellent", new LeftLinearFuzzySet(3.65, 3.875));
Core.addTerm("Good", new TrapezoidFuzzySet(2.9,3.0,3.5,3.875));
Core.addTerm("Poor", new RightLinearFuzzySet(2.5,2.9));
```

The types of fuzzy sets can be trapezoidal, triangular, singleton, PI or many others depending on the nature of the data. Since all three subject groups are evaluated in the same manner, the membership functions are defined using the same figures as the performance benchmark. Figure 3.7 shows the graph of the membership functions for fuzzy variable Major with the linguistic terms Excellent, Good and Poor respectively. As the value of membership function reaches 1, the higher the attachment of the linguistic terms are. For example, for Poor linguistic term in Figure 3.7, the fuzzy area between fuzzy values 2.5 and 2.9 falls in between membership function values 0 and 1. Any linguistic values being under 2.5 are considered as completely poor, where there is no sense of fuzziness attached to the value. As for linguistic values being more than 2.9, the fuzziness for the next linguistic term which is Average is calculated. The scenario is similar for all the linguistic terms. Commonly, the right and left points of the graph usually form a constant line at 1 or 0.

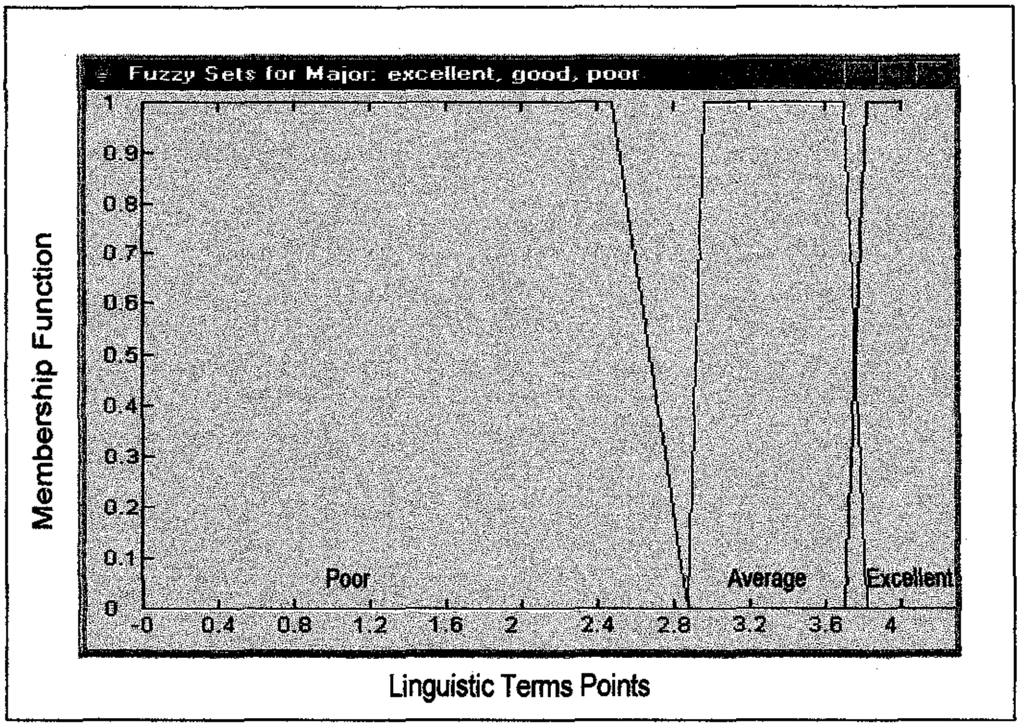


FIGURE 3.7: Membership function for fuzzy variable Major.

3.1.5.2 Construction of fuzzy rules, aggregation and defuzzification

Heuristically induced, there are six final fuzzy rules that are applied by the system. One of the parameters that are used to implement the fuzzy logic system is union function; where it is used to globalize or aggregate the executed rules. The steps require the individual rules to be executed prior to the aggregation of the fuzzy outcome. If the rule antecedents match the rule inputs, rule will fire with meaningful outputs. The test is that all of the antecedent/input pairs overlap at least to degree as specified by the threshold that is set to zero. A global fuzzy value is created to unionize the rule execution outcome prior to defuzzification process. In summary, the steps that are applied in constructing fuzzy rules, aggregation and defuzzification are: (1) Apply the inputs to all of the rules in the system, executing the rules one at a time and performing a global accumulation of the outputs, (2) defuzzify the outputs to create crisp numbers from the output fuzzy values, and (3) apply the crisp outputs to the system. There are two different defuzzifiers that are used in this project:

- Center of Area (COA) defuzzification

This method defuzzifies a fuzzy set returning a floating point that represents the fuzzy set. It calculates the x value that splits the fuzzy set so that there is an equal area on either side of the x value. The set is subdivided into different shapes by partitioning vertically at each point in the set, resulting in rectangles, triangles, and trapezoids.

- Moment defuzzification

This method defuzzifies a fuzzy set returning a floating point that represents the fuzzy set. It calculates the first moment of area of a fuzzy set about the y axis. The set is subdivided into different shapes by partitioning vertically at each point in the set, resulting in rectangles, triangles, and trapezoids. The centre of gravity (moment) and area of each subdivision is calculated using the appropriate formulas for each shape.

The defuzzification process is then followed by the iteration of all Student data. Finally, the accuracy level is calculated for classes accordingly prior to combining the total errors for all instances.

3.2 Unit and Integration Testing

Unit testing involves verifying each module as to determine if it meets the stated specifications. In this project, the accuracy level of individual components in the fuzzy system and the neural network are tested during design phase. The design will then go through integration testing where several modules are combined. The individual components previously mentioned are:

- Fuzzy system.
- Neural network.

- Bayes probability calculations.

Following the success of these modules, the integration test begins through combination of module incrementally. Any incompatibility discovered at this stage is debugged by modifying the linkage between the modules or generally viewing the system as a solid entity without having to debug the modules' internal contents.

3.3 Required Tools

Since the main utilized programming language in this project is Java, the main requirement is Java Runtime Environment, with additional complementary Java tools. The data is to be stored in mySQL5-based database. Other than that, publicly available training and test data sets are also required to train system.

CHAPTER 4

RESULTS AND DISCUSSION

4 RESULTS AND DISCUSSION

After executing both techniques on both data sets, the summary of the results in terms of accuracy is as shown in Table 4.1. Among the three techniques, the highest accuracy is achieved by the combination of back propagation feed forward neural network with initialization of weights using Bayes Theorem. The test is conducted for 20-folds using randomized data selection as the folding method.

	Accuracy(%)			
	NN		NN+B	
Data Set	Train	Test	Train	Test
Iris	89.5-96.2	89.2-96.0	90.1-97.9	90.3-98.6
Student	88.2-90.0	87.0-90.0	88.6-96.3	88.1-97.0
Averaged Iris	92.85	92.6	94	94.45
Averaged Student	89.2	88.5	92.45	92.55

	Accuracy(%)	
	FL	
Data Set	MomentDefuzzify	COA
Iris	80.1	66.0
Student	99.5	75.0

TABLE 4.1: Level of accuracy for neural net and fuzzy system

For Fuzzy Logic, since it is not a learning technique on its own, the defuzzifier with the highest accuracy level returned is moment defuzzifying technique. It is concluded that the results generated by the combination of neural network and Bayes Theorem are higher as compared to Fuzzy Techniques.

For neural network, the error rates are minimized as the iteration increases. Based on the parameters provided in the methodology section of this report, the application is developed utilizing the neural network with initialization of weights using Bayes Theory.

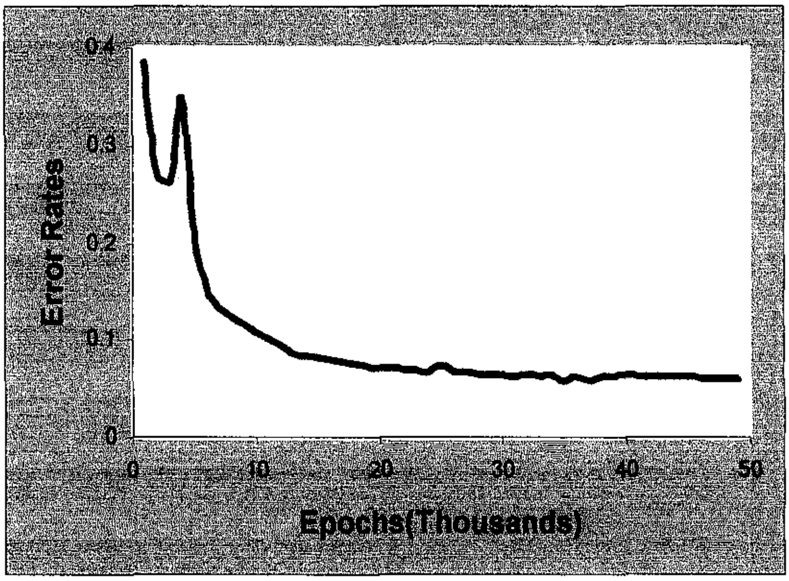


FIGURE 4.1: Error rates for neural network with Bayes weight initialization

For implementation purposes, the chosen methodology is the combination of Bayes and neural network. The ability of neural network to process and learn is much appreciated in a data mining project such as this. Fuzzy technique is used here to present the data in simple human language instead of statistics and numbers. Observing from the two sub-projects that are managed separately, both techniques can function independently for different advantages. In the future, the combination of these techniques may lead to better results in terms of accumulating learning and presentation features.

4.1 Screen Flow

The system is fairly simple for front-end users. User logs on the system, fill in username and password to enter the screen provided in Figure 4.1. Next, student ID is inserted into the text field to be searched. Once found in the database, the student's expected class of graduation will be displayed. As seen in Figure 4.1, the fuzzy table indicating fuzzy rules that are relevant to the classification is included for reference. User can also train the data through front-end access to view the generated statistics and graphs in command prompt format.

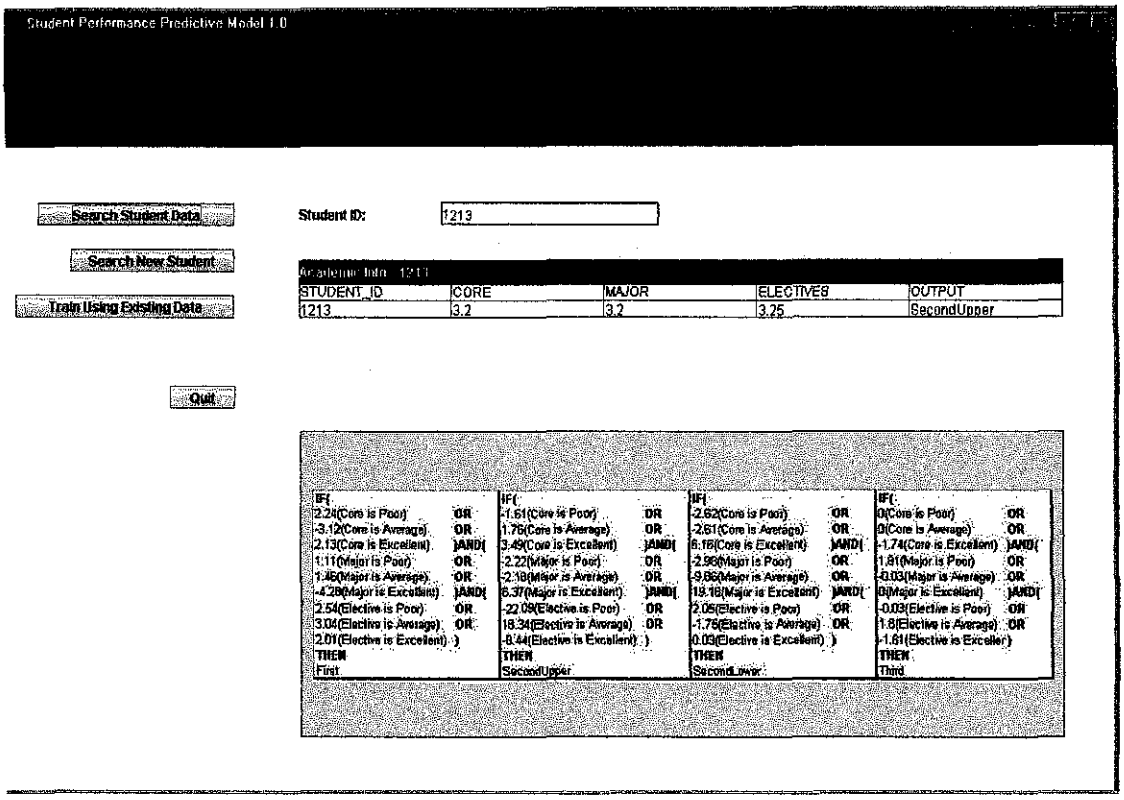


FIGURE 4.2: Student Predicted Class of Graduation

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5 CONCLUSION AND RECOMMENDATION

Fuzzy logic, neural network and Bayes individually covers many areas of possibilities. Although in this project, the techniques are separated, it is foreseen that through the combination of all three techniques, the accuracy level of the returned results is expected to be optimized. Although nowadays most researchers are aiming for hybrid systems, the application of individual fuzzy techniques, neural network and Bayes Theorem should not be left out. Many parties believe that the studies on individual algorithms, and their variants of application are not conducted vastly, thus disallowing direct comparison of algorithms. As further standards are set, the comparison may stand of more solid grounds, and the 'best' algorithm may be named depending on factors accordingly. Thus, the optimization of the subcomponents should be researched on continuously. Through the identification of weaknesses as strengths of the components, the overall objective can be achieved through the construction of an ideal hybrid system.

In specific reference to student academic data, this project has contributed in terms of customizing the grading system based on UTP academic structure. Researchers [24], [25], [8] have identified the problems of grading in general. Tremendous efforts had to be put into the pre-processing stage, where extensive data cleaning and patching was manually conducted. This grading system is generally relevant to many universities in Malaysia as the grading system is fairly similar. However, customization may be needed in order to cater for very specific requirements. It is a possibility that some of the applied methods by these researchers can be absorbed into UTP's way of grading, if not all local universities in Malaysia.

Initial efforts have also been shown in terms of combining learning abilities of neural network and the simple human language presentation of information. In the future, this effort can be continued along with deeper analysis on how Bayes Theorem can further improve the intelligent system as a whole. Other than that, UTP raw academic data's pre-processing should be automated, especially in terms of data cleaning and patching. It is more organized thus definitely is adding to UTP's value in its academic data operations. It is also easier in terms of conducting analyses and knowledge discovery. In conclusion, the system is currently developed at its initial stage, where further technical improvements and combinations of algorithms can be explored to create a more solid system as a whole.

6 REFERENCES

[1] Frawley W., Piatetsky-Shapiro G., Matheus C., 1992, "*Knowledge Discovery in Databases: An Overview*" AI Magazine Fall, Page(s):213-228.

[2] "*NCSA Data Mining*", D2K Toolkit User Manual <http://alg.ncsa.uiuc.edu/tools/docs/d2k/manual/dataMining.html> (accessed March 2006)

[3] Fayyad U., Piatetsky-Shapiro G., Smyth P., 1996, "*From Data Mining to Knowledge Discovery in Databases*", American Association for Artificial Intelligence.

[4] "*Classification and Regression Trees (C&RT)*", 1984, Electronic TextBook StatSoft, STATISTICA <http://www.statsoft.com/textbook/stcart.html> (accessed March 2006)

[5] Jiang L., Zhang H., 2005, "*Learning instance greedily cloning naive Bayes for ranking*", Fifth IEEE International Conference on 27-30 Nov. 2005 Page(s):8.

[6] Breiman L., Freidman J., Olshen R., Stone C., 1984, "*Classification and Regression Trees*". Wadsworth, Monterey, CA.

[7] Stergiou C., Siganos D., "*Neural Network.*". Journal Vol.14 IC, United Kingdom.

[8] Rasmani K. A., 2002, "*A Data-Driven Fuzzy Rule-Based Approach for Student Academic Performance Evaluation.*", Centre for Intelligent Systems and their Applications School of Informatics, University of Edinburgh

[9] Haskell R. E., 1998, *"Neuro-Fuzzy Classification and Regression Trees."*, Proc. Third International Conference on Applications of Fuzzy Systems and Soft Computing, Wiesbaden, Germany, October 5-7.

[10] Lewis R. J., 2000, *"An Introduction to Classification and Regression Tree (CART) Analysis."*, Harbor-UCLA Medical Center California, Presented at the 2000 Annual Meeting of the Society for Academic Emergency Medicine.

[11] Bittencourt H. R., Clarke R. T., 2004, *"Feature Selection by Using Classification and Regression Trees(CART)"*, Congress, 12-23 July Turkey Faculty of Mathematics Pontifícia Universidade Católica do RS Brazil.

[12] Dubois D. and Prade, H., 1996, *"What Are Fuzzy Rules and How to Use Them"*, Fuzzy Sets and Systems, Page(s):84.

[13] Mendel J.M., 2001, *"Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions."*, Prentice Hall PTR, New Jersey.

[14] Stutz J., Cheeseman P., 1994, *"A Short Exposition on Bayesian Inference and Probability."*, National Aeronautic and Space Administration Ames Research Centre: Computational Sciences Division, Data Learning Group.

[15] Chun C.F., Iyer V., Brown W., Kok W.W., 2005, *"Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity Machine Learning and Cybernetics."*, Proceedings of 2005 International Conference on Volume 1, 18-21 Aug. 2005 Page(s):394 – 398.

[16] Liang C., 2005, *"Pattern classification by assembling small neural networks"*, IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on Volume 3, Page(s):1947 – 1952.

[17] Holst A., 1997, "*The Use of a Bayesian Neural Network Model for Classification Tasks.*", Dissertation, Department of Numerical Analysis and Computing Science Royal Institute of Technology, Sweden.

[18] Zheng J., Tang Y., 2004, "*Fuzzy Modeling Incorporated with Fuzzy D-S Theory and Fuzzy Naive Bayes*", Lecture Notes in Computer Science, Springer Berlin / Heidelberg on Volume 3339, Page(s):816.

[19] Do Wan Kim, Jin Bae Park, Young Hoon Joo, "*Fuzzy Classifier with Bayes Rule Consequent*", Lecture Notes in Computer Science, Springer Berlin / Heidelberg on Volume 3809, Page(s):1130 – 1133.

[20] Nurnberger, A., Borgelt, C., Klose, A., 1999, "*Improving Naive Bayes Classifiers Using Neuro-Fuzzy Learning*", Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on Volume 1, Page(s): 154-159.

[21] Zhang, H., Shengli Sheng, 2004, "*Learning weighted naive Bayes with accurate ranking*", Proceedings. Fourth IEEE International Conference on 1-4 Nov. 2004. Page(s):567 – 570.

[22] Orchard, R., 2001, "*Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and FuzzyJess*", Proceedings of the ICEIS 2001, Third International Conference on Enterprise Information Systems, Setubal, Portugal. July 7-10, 2001. Page(s): 533-542.

[23] Marin-Blazquez, J.G. and Shen, Q., 2002. "*From Approximate to Descriptive Fuzzy Classifiers*", IEEE Transactions on Fuzzy Systems, Volume 10 Issue 4, 2002 Page(s):484 – 497.

[24] Rasmani K.A. and Shen Q., 2005, "*Subsethood-based Fuzzy Rule Models and Their Application to Student Performance Classification*", IEEE International Conference on Fuzzy Systems.

[25] Rasmani K.A and Shen Q., 2004, *"Modifying Weighted Fuzzy Subsethood-based Rule Models with Fuzzy Quantifiers"*, IEEE International Conference on Fuzzy Systems.

[26] Echauz J.R. and Vachtsevanos G.J., 1995, *"Fuzzy Grading System"*, IEEE Transactions on Education.

Student Test Data Set

ID	Core	Major	Elective	Class
1418	3.79	3.89	4	First
1420	3.67	3.78	3.75	First
1427	3.71	3.89	4	First
1428	3.59	3.78	3.75	First
1431	3.75	3.89	4	First
1432	3.66	3.78	3.75	First
1433	3.68	3.76	3.74	First
1434	3.9	3.65	3.78	First
1413	2.9	2.8	2	SecondLower
1416	2.5	2.5	3.05	SecondLower
1417	2.9	2.99	3.1	SecondLower
1422	2.9	2.8	2.5	SecondLower
1425	2	2.58	3.05	SecondLower
1426	2.9	2.8	3.1	SecondLower
1429	2.4	2.58	3.05	SecondLower
1430	3	2.8	2.9	SecondLower
1411	3.5	3.15	3	SecondUpper
1412	3.1	3	3.05	SecondUpper
1421	3.15	3.2	3.49	SecondUpper
1435	3.55	3.4	3.3	SecondUpper
1436	3.1	3.44	3.19	SecondUpper
1437	3.09	3.5	3.11	SecondUpper
1438	3.44	3.21	3.29	SecondUpper
1439	3.59	3	2.9	SecondUpper
1414	2	2	1.87	Third
1415	2	1.5	1.59	Third
1419	2	2.3	1.99	Third
1423	2.5	2	1.98	Third
1424	2.1	2	2.05	Third
1440	2.5	2.3	2	Third
1441	2.4	2.5	2.01	Third
1442	1.99	2.51	2	Third

APPENDIX 3-2

Student Raw Data Sample

ID	Semester	Program	Course Code	Grade	Course Code	Grade	Course Code	Grade
1	Jul-04	CO03	SLB1063	B+	SFB3023	B	STB5023	B
1	Jan-04	CO03	STB4123	D+	STB4083	C+	STB4043	B+
1	Jan-03	CO03	STB4013	A-	STB4073	B	STB4093	D+
1	Jan-05	CO03	STB5033	B+				
1	Jan-02	CO03	STB2073	C+	STB2123	C+	STB3013	B
2	Jul-04	CO03	SSB5073	A-	STB5013	B+	STB5023	B+
2	Jan-04	CO03	STB5012	C+	SNB3023	B	STB4033	B
2	Jan-03	CO03	STB4013	A	STB4073	B	STB4093	B
2	Jan-02	CO03	STB2073	B+	STB2123	B	STB3013	A-

Course Code	Grade	Course Code	Grade	Course Code	Grade	GPA (C)	CGPA (C)
STB5013	C+	SSB5073	B			**3.00**	2.72
STB4033	B	STB5012	C+	SFB3013	C	**2.50**	4.00
SLB3013	C+	SFB1013	C	SNB4023	C+	**2.54**	2.50
						3.50	2.73
STB3043	C+	SLB1023	C+	KKB2011	P	2.6	3.05
STB5033	B	SNB3033	A-			**3.50**	3.36
STB4043	A-	STB4083	A-	STB4123	B	**3.21**	4.00
SFB1013	B	SWB1013	C+	SNB4023	B	**3.08**	2.75
STB3043	A-	SLB3013	B+	KGB1041	P	3.5	3.71

GPA (P)	CGPA (P)	ACH	Year	Prev. sem. Standing	Current Standing
2.50	2.68	**153**	4		GOOD STANDING
2.68	2.72	138	4		PASS
2.51	2.51	102	3		PASS
3.00	2.72	156	4		GOOD STANDING
2.51	2.49	66			PASS
3.21	3.35	**153**	4		DEAN'S LIST
3.35	3.37	138	4		GOOD STANDING
3.27	3.31	105	4		GOOD STANDING
3.46	3.44	69			DEAN'S LIST