# Classifying Student Academic Performance: A Hybrid Approach

Ahmad I. Z. Abidin, Ili A. Setu, S. P. Yong, O. M. Foong, Jale Ahmad

*Abstract*—**Nowadays, organizations are overwhelmed with a large amount of electronic data that require proper management to discover previously hidden knowledge. Having a set of non-transformed data may be a huge waste as specific processes onto the data would result in the discovery of valuable knowledge. This paper describes the development of a predictive model to classify undergraduate students' class of graduation: first class, second class upper division, second class lower division, or third class. The hybrid techniques used to support the classification are Bayesian probability and neural network.**

*Index Terms*—**Bayesian probability, neural network, data mining, artificial intelligence**

## I. Introduction

Data Mining (DM) is the non-trivial of implicit, previously unknown, and potentially useful information from available data [1]. DM has been widely used in many business sectors. An example of DM application is in marketing in which customers' profile classification is performed based on demographic data. In addition to business sectors, DM can also be practical to track students' performances.

In this paper, the research aims at analyzing Bayesian probability and neural network utilization on the prediction accuracy of a student's class of graduation. Identifying each student's areas of strengths and weaknesses is a challenge. Through the application of DM methods, a student' academic performance can be forecasted by analyzing the strengths and weaknesses of all courses required in the student's field of study.

The following two scenarios highlight the importance of the research. Firstly, an academic performance classifier can be a tool to help high achievers maintain their excellence. Secondly, the classifier addresses on students requiring urgent attention for academic rehabilitation. It is a common practice that a student will be placed under probation if the student's CGPA or Grade Point Average (GPA) is not at par with the university's minimum requirements. In this case, the analysis narrowly focuses on determining the best solution to improve academic performance to avoid dismissal. Currently, the academic rehabilitation session mostly involves heuristics-based decisions mutually agreed by both students and lecturers who also act as academic advisors.

The objective of this paper is to demonstrate that the utilization of neural network and Bayesian probability in classification and prediction of student academic performance returns a relatively good accuracy level. Sample dataset of undergraduates of Universiti Teknologi PETRONAS (UTP) were applied.

## II. Related Work

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]. The knowledge discovery goals are defined by the intended use of the system. There are two types of goals: verification and discovery. With verification, the system is limited to verifying the user's hypothesis, whereas 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 paper both predictive and descriptive KDD are used.

### A. Predictive and Descriptive KDD

#### 1) Data Selection and Data Cleaning

Relevant data is selected from data warehouse or legacy system. Cleaning of data formats, merging of data and removal of trailing comments are performed prior to transformation process.

#### 2) Data Transformation

The extract, transform, and load procedures are performed at this stage where desired data in specific format are gathered in a database known as the data mart.

#### 3) Data Mining / Pattern Discovery

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

#### 4) Interpretation and Evaluation

Knowledge obtained from Data Mining is interpreted and evaluated.

### B. Artificial Neural Network (ANN)

An artificial neuron has input and output. The neuron has training and testing modes of operation. In the training mode,

Manuscript received November 20, 2007. This work was supported in part by Universiti Teknologi PETRONAS. Ahmad I. Z. Abidin is the corresponding author and attached to the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, MALAYSIA (tel: +605-368-7487; Fax: +605-365-6180; e-mail: izuddin_z@petronas.com.my).

Ili A. Setu graduated with Bachelor of Technology (Hons.) degree in Business Information System, Universiti Teknologi PETRONAS.

S. P Yong, O. M. Foong and Jale Ahmad are with the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, MALAYSIA. They can be contacted via email at {yongsuetpeng, foongoimean, jale_ahmad}@petronas.com.my.

the neuron can be trained to fire a particular input pattern. Then, in the testing 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 [4].

Information is stored in the weight matrix *W* of a neural network. Two major categories of neural network are fixed network and adaptive network. In fixed network, weights are non-changeable. In such network, the weights are fixed according to the problem to solve. The adaptive networks, on the other hand, are able to change their weights.

There are two types of ANN architecture. Firstly, a feed forward ANN allows signals to travel one way only, from input to output. The feed forward ANN tends to be straightforward, associating inputs with outputs. It is extensively utilized in pattern recognition. Secondly, a feed back network can have signals traveling in both directions by introducing loops in the network. The feed back network is also referred to as interactive or recurrent, although the latter term is often used to denote feed back connections in single-layer organizations. [4]

One of the most popular neural network architectures used for classification is the Multi-Layer Perceptron (MLP). The units are organized into different layers, and the network is a 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 MLP 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, propagated through the network to the output layer. The output of the network is then compared to the desired output, and the difference gives rise to an error signal which is fed backward through the network, causing the weights to be updated in a way which will decrease the error the next time the same pattern is discovered. By going through the training set several times, the weights are gradually adjusted to minimize the output error [5].

### *C. Statistical Method: Bayesian Theorem*

Bayes' Theorem, developed by the Rev. Thomas Bayes, an 18th century mathematician and theologian, was first published in 1763 [6]. The probability is calculated where a Bayesian inference can be made based on available information. There are two major classifications using different Bayesian rules [5]: (i) one-layer neural network to deduce Bayesian learning rule and (ii) multi-layer neural network having complex columns in hidden layer. 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 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 Naïve 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 the already activated units. Related work can be broadly divided into two approaches: eager learning and lazy learning, depending on when the major computation occurs. The eager learning performs major computation at training time. The lazy learning, on the other hand, spends little or no effort during training and delaying computation until classification time [7]. Interested readers may find more information in neural network and Bayesian technique in [8-12].

## III. SYSTEM DESIGN

Back propagation feed forward neural network (BPFFNN) is utilized to observe the accuracy level returned on the dataset. Fig. 1 shows the system design in a more comprehensible structure. There are two independent sub-systems, *subA* and *subB*, executed in a parallel manner. *SubA* assigns random weight ranging from 0.00 to 4.00 to the neural network. The random value actually signifies minimum and maximum value of a GPA whereas *SubB* initializes weight using Bayesian probability. The purpose of developing sub-system *subA* is to compare the accuracy of its prediction results with the prediction results sub-system *subB* yields.

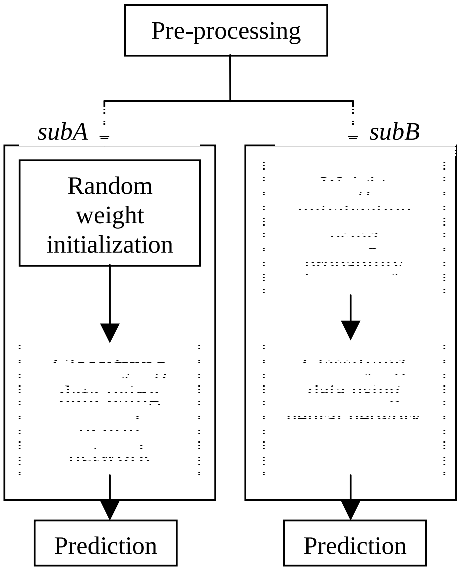
Pre-processing
subA
subB
Random weight initialization
Weight initialization using probability
Classifying data using neural network
Classifying data using neural network
Prediction
Prediction

Fig. 1: System design of the proposed application

### *A. Pre-processing*

The pre-processing procedure applies for the raw academic data obtained from the Academic Central Services of UTP. Publicly available test datasets usually require no pre-process as they have been prepared for testing purposes. The pre-processing is to produce datasets of the desired format. In UTP, student data require extensive patching processes due to factors such as change of curriculum structures and special circumstances that leads to outliers. For example, a student who changes academic program during his/her duration of studies may have additional subjects in record. Currently, extraction and transformation processes are not fully automated as data cleaning needs massive human intervention to allow correct data interpretation.

Currently, there is no formal system in UTP used to identify a student's academic status. The academic data go through the process of producing current results in specific values of GPAs and CGPAs. No prediction or mining is currently conducted in identifying strengths or in avoiding dismissals. The retrieval of raw academic data is followed by filtering, cleaning and patching up process. The initially provided instances total up to 400, but the pre-processing stage filters out invalid data to produce 300 instances ready for testing. The student raw academic data consist of course, grade, GPA, and CGPA columns. These data are grouped into three specific fields: (1) Core, (2) Major, (3) Electives. The grades for respective subjects are then averaged as grouped by these three fields. The dataset referred in this paper is named Student. Table I provides the attributes of the Student dataset.

Table I: A summary of Student dataset

| **Dataset** | Student |
|---|---|
| **No of instances** | 300 |
| **No of input** | 3 |
| **No of output** | 4 |
| **Others** | CGPA mean of three subject groups |

Following pre-processing stage, the dataset consists of 300 instances, and three linguistic variables: excellent, average, and poor. The output is first class, second class upper division, second class lower division, or third class. Generally, at UTP, the mapping of CGPA to class of graduation is as shown in Table II.

Table II: Mapping of CGPA to class of graduation

| **CGPA range** | **Class of graduation** |
|---|---|
| 3.75 – 4.00 | First class |
| 3.00 – 3.74 | Second class upper division |
| 2.50 – 2.99 | Second class lower division |
| 2.00 – 2.49 | Third class |

To pre-process the raw student dataset, basic database operations are conducted. Each subject is tagged with a subject group; being either core, elective or major. The following formula is applied on all instances to get the Bayesian probability of each feature:

$$P(H\,|\,E,c) = \frac{P(H\,|\,c) * P(E\,|\,H,c)}{P(E\,|\,c)} \tag{1}$$

$P(H\,|\,E,c)$ is the probability of $H$ after considering the effect of $E$ on condition $c$. The term $P(H\,|\,c)$ is called the prior probability $H$ of given $c$ alone. The term $P(E\,|\,H,c)$ is the likelihood and gives the probability of the evidence with the assumption that the hypothesis $H$ and the background information $c$ are true. Finally, the term $P(E\,|\,c)$ is independent and can be regarded as a normalizing or scaling factor.

Thirty six weight values from the matrix are generated, noted as W1, W2,…, and W36. Initially, calculations are conducted based on features and classes, and next combining the values into sub-results of the nominator and denominator.

There are four functions developed to calculate the Bayesian probability. The purposes of the functions are to calculate set probability sub-values and to get the probability sub-values for final calculation. The Bayesian probability values are then mapped to the number of neurons in the neural network hidden layer 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 assigned to each possible combination of linguistic variables and linguistic terms. Table III shows the structure of Student dataset for Bayesian probability calculation.

Table III: The structure of student dataset for Bayesian probability calculation

| **Variable input** | **Linguistic term** | **Class** | | | |
|---|---|---|---|---|---|
| | | **Third** | **Second Lower** | **Second Upper** | **First** |
| Core | Poor | W1 * CP | W10 * CP | W19 * CP | W28 * CP |
| Core | Average | W2 * CA | W11 * CA | W20 * CA | W29 * CA |
| Core | Excellent | W3 * CE | W12 * CE | W21 * CE | W30 * CE |
| Major | Poor | W4 * MP | W13 * MP | W22 * MP | W31 * MP |
| Major | Average | W5 * MA | W14 * MA | W23 * MA | W32 * MA |
| Major | Excellent | W6 * ME | W15 * ME | W24 * ME | W33 * ME |
| Elective | Poor | W7 * EP | W16 * EP | W25 * EP | W34 * EP |
| Elective | Average | W8 * EA | W17 * EA | W26 * EA | W35 * EA |
| Elective | Excellent | W9 * EE | W18 * EE | W27 * EE | W36 * EE |

In Table III, CP stands for core is poor, CA core is average, and CE core is excellent. MP stands for major are poor, MA major is average, and ME major is excellent. EP stands for elective is poor, EA elective is average, and EE elective is excellent. W1 – W9 correspond to third class, W10 – W18 to second class lower division, W19 – W27 to second class upper division, and W28 – W36 to first class.

A two-layer of hidden network is used in BPFFNN as shown in Fig. 2. H1 – H7 represent the hidden nodes. The activation function used is Sigmoid-based. The BPFFNN includes three inputs, two hidden layers and four outputs. Weights derived from Bayesian probability are assigned to each synapse that connects the neurons. These weights are trained until the error is minimized. The learning process for the BPFFNN 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. There are two conditions that halt the training: when the maximum number of epochs equals to 50000 or error ratio is less than 0.05.

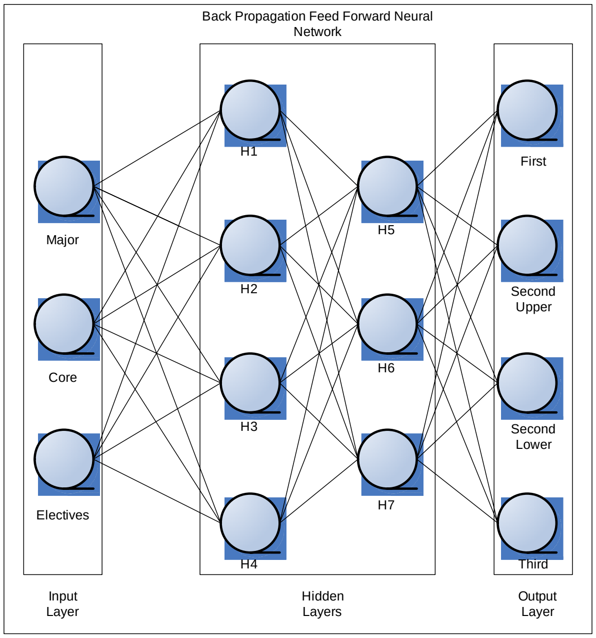

Fig. 2: Architecture of BPFFNN for student dataset

## IV. Results and Discussions

The summary of the results is as shown in Table IV.

Table IV: Level of accuracy for Neural Network (NN) and Hybrid system (B+NN)

| | Accuracy (%) | | | |
|---|---|---|---|---|
| | **NN** | | **Hybrid (B+NN)** | |
| **Dataset** | **Train** | **Test** | **Train** | **Test** |
| Student | 88.2-90.0 | 87.0-90.0 | 88.6-96.3 | 88.1-97.0 |
| Average | 89.2 | 88.5 | 92.45 | 92.55 |

The accuracy of both sub-systems was calculated based on predicting the outcome of class of graduation correctly. On average, the accuracy achieved is higher for the combination of neural network with initialization of weights using Bayesian probability (B+NN) than for neural network (NN) with random weights initialization on both training and testing data. The test is conducted for 20-folds using randomized data selection as the folding method.

The accuracy of classification for training data using NN ranges from 88.2 – 90.0%, whereas the accuracy of prediction for testing data using NN ranges from 87.0 – 90.0%. The accuracy of classification for training data using B+NN ranges from 86.6 – 96.3%, whereas the accuracy of prediction for testing data using B+NN ranges from 88.1 – 97.0%.

Based on the Student dataset, the hybrid approach which combines Bayesian probability and neural network, on average, shows improvement over an approach using neural network with random weight assignment.

## V. Conclusion

Neural network and Bayesian theorem individually cover many areas of possibilities. In specific reference to student academic data, this research has contributed in building initial predictive model for academic performance based on UTP academic structure. The developed predictive model can also be applied to students' data of other universities that sort the class of graduation of their students.

In the future, this effort shall continue along with in-depth analysis on how Bayesian probability can further improve the system. Besides, UTP raw academic pre-processing data should be auto-cleaned and auto-patched. Further technical improvements and combinations of algorithms can be explored to create a more solid system.

## Acknowledgment

The authors would like to thank the Academic Central Services of Universiti Teknologi PETRONAS for providing data useful in this research.

## References

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

[2] Cios**,** K.J., Pedrycz**,** W., Swiniarski**,** R.W., Kurgan**,** L.A. "Data Mining: A Knowledge Discovery Approach", Springer, 2007.

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

[4] Stergiou C., Siganos D., "*Neural Network.*" Surprise 96 Journal Vol. 4, United Kingdom, 1996.

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

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

[7] Jiang L., Zhang H. "*Learning instance greedily cloning naïve Bayes for ranking*", Proceedings of Fifth IEEE International Conference on Data Mining, 27-30 Nov. 2005, pp 202-209.

[8] Liang C., 2005, "*Pattern classification by assembling small neural networks*". Proceedings of IEEE International Joint Conference on Neural Networks, Vol. 3, 2005, pp 1947 – 1952.

[9] Zheng J., Tang Y. "*Fuzzy Modeling Incorporated with Fuzzy D-S Theory and Fuzzy Naive Bayes*", Lecture Notes in Computer Science, Springer Berlin / Heidelberg on Vol. 3339, 2004, pp 816.

[10] 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, pp 1130 – 1133.

[11] Nurnberger, A., Borgelt, C., Klose, A. "*Improving Naive Bayes Classifiers Using Neuro-Fuzzy Learning*". Proceedings of 6th International Conference on Neural Information Processing, Vol. 1, 1999, pp 154-159.

[12] Zhang, H., Shengli Sheng. "*Learning weighted naive Bayes with accurate ranking*". Proceedings of Fourth IEEE International Conference on Data Mining, 1-4 Nov. 2004, pp 567 – 570.