Using an Enhanced Feed-Forward Neural Network Technique for Prediction of Students' Performance

Adeleke Ajiboye, Ruzaini Abdullah-Arshah, and Hongwu Qin

***Abstract*—** The newly admitted students for the undergraduate programmes in the institutions of higher learning sometimes experience some academic adjustment that is associated with stress; many factors have been attributed to this, which most times, results in the high percentage of failure and low Grade Point Average (GPA). Computing the earlier academic achievements for these sets of students would make one to be abreast of their level of knowledge academically, in order to be well-informed of their areas of weakness and strength. In this paper, an enhancement of Feed-forward Neural Network for the creation of a network model to predict the students' performance based on their historical data is proposed. In the course of experimentations with Matlab software, two network models are created using the existing and enhanced feed-forward neural network techniques. The ability of these models to generalize is measured using simulation methods. The enhanced network model consistently shows a high degree of accuracy and predicts well. The performance of students predicted as outstanding, can also be supported financially in the form of scholarship; while those that are found to be academically weak can be encouraged and rightly counseled at the early stage of their studies.

***Keywords*—** data partitioning, neural networks, predictive model, students' performance

I. Introduction

THE technique of data mining is one of the established methods that can be used to reveal knowledge from data through the construction of models. Traditional data analysis techniques make predictions about the future based on a sequence of rules generated from past data; it creates classifications based on these rules which contain empirical knowledge [1]. The use of neural network technique is relatively different from the process involved in statistics, as the technique does not require identifying empirical rules in order to make predictions. Instead, a neural network generates a network model by mapping all the significant patterns and relationships that exist among specified predictive attributes, otherwise referred here as input. In a supervised learning, the network uses another specified attribute usually referred to as target to predict an output

The technique of feed-forward neural network requires back propagation algorithm for training and from the statistical point of view, given enough hidden units and sufficient training samples, multi-layered feed-forward network can closely approximate any function [2]. This network is the most popular neural network architecture and the network is known for its analytical tractability and effectiveness [3]. Enhancement of this network technique is, therefore, inevitable to sustain its performance.

One of the useful indicator to measure the network's performance is the sum of square errors [4]; other common performance measurement for numeric predictions are: mean absolute error, mean square error, relative square error and correlation coefficient [5]. The way data is partitioned also play important roles in network generalization, essentially, training data are expected to cover the full range of the input space [6]. Earlier studies have also revealed some effective ways by which errors can be significantly reduced in a network during training. For instance, as proposed in [7], the study showed that the error rate can be reduced by continuously adjusting the weights of network connections, as this would minimize the difference between the actual vector output of the network and the target output vector.

The present paper presents a technique for improving the performance of feed-forward neural network for efficient exploration of educational data in order to achieve predictions of high accuracy and better generalization. We organized the rest of the paper as follows: In the next section, the concept of modelling using neural network technique is discussed. In section 3, some related works reported in the literature are discussed. Section 4 discusses the proposed approach and in section 5, the analysis of the results of this study based on the enhanced and existing network models is presented. The study is concluded in section 6.

II. Neural Network Modeling Techniques

A neural network is a massively parallel distributed processor, made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use [8]. It resembles the brain in two respects: knowledge is acquired by the network from its environment through a learning process; also, interneuron connection strengths, known as synaptic weights, are used to

Adeleke Ajiboye is a PhD candidate at the Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, 26300 Gambang, Malaysia (phone: +601116214234; e-mail: ajibraheem@live.com).

Ruzaini Abdullah-Arshah is an Associate Professor, he is with the Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, 26300 Gambang, Malaysia (e-mail: ruzaini@ump.edu.my).

Hongwu Qin is a Senior Lecturer with the Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, 26300 Gambang, Malaysia (e-mail: qin.hongwu@ump.edu.my).

store the acquired knowledge.

Feed-forward neural network is a perceptron network with one or more hidden layers, each network edge represents a scalar information flow, so that the number of input neurons, p, and the number of output neurons q specifies the input and output dimensionality of the network, which then realizes a function with p inputs and q outputs, f: RP → Rq. A common training process for the purpose of learning in feed-forward neural networks is the back propagation process [9]. The techniques of neural networks can learn new associations, patterns and functional dependencies; the learning changes the network's memory either by updating its status or by adding new facts and since the neural network does not use a mathematical model of how a system's output depends on its input, they behave as model-free estimates [1].

III. Related Work

In this section, some related work on enhancing the performance of neural network and using neural network for predictive model construction is discussed. Improving the neural network performance in order to achieve a more accurate prediction has been of interest to many researchers. Some of the earlier works proposed in this area included: improving how the neural network performs on new inputs [10], improving the topology of the network [11], enhancement on algorithms for network training [12] etc.

Designing the neural network architecture for a satisfactory performance is a complex task; this most times involve using several numbers of neurons in the hidden layer(s), however, the use of pruning algorithm proposed in [13] and constructive algorithms for structure learning in feed-forward neural network proposed in [14] provides some elaborate methods.

The performance of feed-forward network is enhanced in [3] using 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 feed-forward network. Earlier studies of supervised learning in a multi-layered feed-forward network [15] revealed that, architecture based on small-world reduces both learning error and training time when compared to regular network.

The use of computer –aided optimal design proposed in [16] is also another method aimed at improving neural network predictive ability. In the study, the approach designed a strategy based on statistical concepts in order to improve the feed-forward network generalization. The study carried out Monte Carlo-simulation based method to examine the usefulness of the design approach in the context of feed-forward network. Study in [17] proposed an enhancement of the feed-forward network 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 study in [18], proposed a modified particle swarm optimization algorithm to select the input weights and hidden biases of single-hidden-layer feed-forward neural networks with a view to improving the predictive ability of the learning machine. The study was reported to have better generalization performance. In the study proposed in [19], the authors argued that the random input weight selection using Extreme Learning Machine (ELM) algorithm may lead to an ill-conditioned problem, which can lead to a solution that 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.

Also, the feed-forward network in its regular structure has been widely used to create a network model for the purpose of making predictions. Study in [20], identified data mining and machine learning as techniques suitable for data exploration, especially to predict the students' academic performance.

The dataset explored in the present work differentiate it from several other related studies reported in the literature. The explored dataset comprised of students' submissions online while seeking admission to undergraduate programmes. The relevant attributes suitable for students' performance prediction are identified to implement the models constructed. We inferred from the reviewed literature that, stored data captured with respect of each registered student after a number of semesters are what is widely reported for the prediction of students' performance. Thus, creating a predictive learning model with the data used in the present study would unveil the background information about the students' achievement. The present study also proposed some enhancement for the existing feed-forward network before it is used to create the learning model in order to ensure better accuracy

IV. The Proposed Approach

In this section, the algorithm used for the partitioning of dataset for learning purposes in the proposed approach is represented in Figure 1, the algorithm is implemented using Matlab software. The study experiments on series of partition options with a view to arriving at the data partitioning that gives optimum results. The study compares the accuracy of each model, especially as relates to how each network model generalizes with the set of new inputs.

Several measures can be used to evaluate the success of numeric predictions, Mean Absolute Error (MAE) is the choice of error measurement to evaluate the accuracy of the simulated outputs of each model in this study. Error computed using MAE does not tend to exaggerate the effect of outliers [5] as all sizes of error are treated evenly according to their magnitude.

```
    // partitioning of dataset for training

Step 1: set data division function to nil        // to off the default data division function
Step 2: assign divide indices to network divide function    \* divideind is the function to be used in lieu of
                                                               dividerand*/
Step 3: set the indices to desired sample size   /* for instance 1 : x;  x+1: y  and  y+1 : z, where z is the size of
                                                    the training data.*/
Step 4: set the network parameters to indices
Step 5: set the divide parameter input indices to corresponding sample input
Step 6: set other network parameters for optimum performance
Step 7: train the network

    // Tracking of error during training using validation set
  N = Number of Epochs      // number of iterations
  n = current iteration
while  n < N       // maximum iteration is set to a particular number (N) say 1000
   {
     If (Error)n+k > (Error)n+k-1   //comparison to check if present error is greater than the previous error
     {
Break...          //stop training if yes, otherwise continue training

     }
   }

End.
```

Fig. 1 Algorithm for the data division and error tracking during training process

4.1 Data Collection and Transformation

In order to implement the proposed algorithm and predictive models constructed in this study, we consider using a real life data sets. Upon request, the dataset was released to us for the purpose of this study at a public university in the north central, Nigeria. The candidates seeking admission for undergraduate programmes were instructed to supply their detailed information through the interface provided on the institution's website. The reason for capturing data in most cases has no bearing on using such data for modelling purposes; therefore, the data initially stored in Mysql in the portal were converted to an Excel file for further processing.

Some of the variables that are considered to be suitable for prediction are identified in these records. The study adopts knowledge-driven approach whereby, the domain knowledge provided by the experts is relied upon. In order to effectively use the technique of data mining to achieve the desired goals, a cooperative effort of humans and computers is inevitable as best results are achieved by balancing the knowledge of human experts in describing problems and goals with the search capabilities of computers [21]. Information received from the Knowledge Domain Experts (KDE) and similar variables that have been reported in the literature guided this study on the choice of attributes, while other variables that are considered to have low predictive relevance were discarded.

The success of data mining technique depends on many factors, study proposed in [22] identified quality and relevance of the data as crucial to model performance, as data sets that is noisy or redundant cannot reveal reliable knowledge. The five selected attributes as shown in Table 1, serves as the inputs, while the overall achievement by each student which serves as a pointer to their academic performance forms the target attribute. The target is normalized to 100% based on (1).

where t = 'target attribute'
x = 'score achieved'
X = 'achievable score'

$$t = \frac{x}{X} \times 100 \qquad (1)$$

TABLE 1
THE PREDICTIVE ATTRIBUTES USED FOR THE MODELLING

Attributes	Coded values		Range
	UTME:		
	200-219	1	1 – 4
	220-229	2	
	230-239	3	
	240 and above	4	
ATR	Direct:		
	ND/NCE/A'LEVEL	2	
	HND	3	
	BSC	4	2-4
	Grades:		
HSR	A1 6; A2, B2 5; A3, B3 4; C4 3 C5 2; C6 1		1-6
AHSE	1 Attempt	2	1-2
	2 Attempts	1	
AGE	14-20	3	1-3
	21-29	2	
	Above 30	1	
	Scores (%)		
	50-59	1	
	60-65	2	1-4
PST	66-69	3	
	70 and above	4	

The aggregate score achieved by each student from these academic/demographic variables is computed. This score serves as the target variable and used together with other five variables, to construct a supervised learning neural network model. The attributes are briefly described:

Admission Type & Result (ATR): The main entry mode to Nigerian university is either through direct (by presenting higher institution result) or by participating in the mandatory examination known as utme (unified tertiary matriculation examination); while the direct entry students join the students in second year at the university, utme students commence their studies in the first year.

High School Result (HSR): In this paper, the quality of high school result is determined and coded in a way that, the highest quality grade achieved in a subject is coded as 6, while the lowest is coded as 1. The West African Examination Council (WAEC) standard is adopted.

*Attempt in High School Examination(AHSE):*Brilliant students sit for examination only once and pass all the subjects while average students may have to attempt such examination one more time in order to achieve the required result for further studies.

Student's Age: The human ability to recall or retain what is stored in their memory, or learn new things a a faster rate, varies with age; under normal condition, younger people are most favoured.

Pre-Admission Screening Test (PST): This is the score obtained by the students in the test conducted by the institution to validate their achievements in ATR.

4.2 Experiments

Experiment 1

In the first experiment, the data is partitioned and trained in conformance with the algorithm shown in Figure 1. The 1300 dataset is partitioned into three parts; the specific percentages of data used for training are: 68, 74 and 80, while the remaining data is divided in equal percentage for the purpose of validation and testing. The repeated experiment shows marginal improvement for the training percentages 68 and 74 as the error rate decreases, while error rises sharply with 80% training set. Specifically, with 74% for training, 13% for validation and 13% for testing, a network model that generalizes better and of high accuracy is achieved in this study. The network configuration (see Figure 2) and architecture of the proposed network model (see Figure 3) is as represented, while defaults are maintained for configuration settings that are not shown.

Algorithm	*Data Division:* Index *Training:* Levenberg Marquardt
Network properties	*Network type:* Feed-forward BP *Performance:* Mean Square Error *Number of Neurons:* 20
Network Parameters	*Epochs:* 750 *Goal:* 0 *Min grad:* 1e-7 *Max_fail:* 12 *Mu* : 0.0001

Fig. 2 Configuration of the feed-forward BP network

The Mean Square Error (MSE) is computed during the training which the validation set monitors to signal the end of training when an increase in error is noticed in order to avoid over fitting. The MSE is computed using the formula in (2).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \qquad (2)$$

where n is the number of samples, is the prediction and is the target value. The MSE thus evaluates the quality of predictions set in terms of its variation and degree of bias.

Experiment 2

In the second experiment, the motive was to use the regular feed-forward to create model using the same dataset and configuration settings used in model construction in experiment 1. The network in its default structure in Matlab implementation partitions dataset to 60% for training, 20% for validation and 20% for testing. Other parameters are left in their default settings

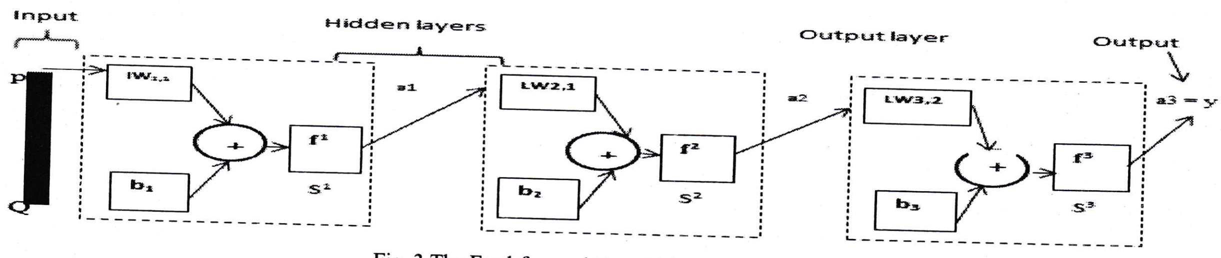

Fig. 3 The Feed-forward Neural Network Architecture

4.3 Accuracy measurement

The network models are evaluated by simulating them using 250 new untrained inputs. This is to determine the performance accuracy of both network models using a dataset that it has not previously seen. The simulation results based on the enhanced feed-forward network is represented in Figure 6, while Figure 5 represents the result of simulating model created using feed-forward network in its regular structure.

The Mean Absolute Error (MAE) is computed on the resulting output for each experiment to determine the pattern of error associated with each dataset division that are used for creating the network model. This is for the purpose of making comparisons and to identify the best partition among the various partition options. The MAE computation is based on (3).

$$\text{MAE} = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{n} \quad (3)$$

The predicted values of the test instances are p1, p2,…., pn; the actual values are a1, a2,…., an. while n is the number of sample dataset.

The neural networks techniques does not generate rules or relationship between the input and output results, but as a machine learning technique, it is able to fit a model from the data. Additional information provided by the knowledge domain expert is shown in Table 2, this is for the purpose of predicting the students' performance based on the network outputs. The table contains the acceptable students' performance information. The achievement by each student as predicted by the network model is therefore, mapped to status and performance for final decision making.

TABLE II
STUDENTS' PERFORMANCE BASED ON KDE

Status	Achievement	Performance
Risk free	75 >= t <=100	Outstanding
Low risk	60 >= t <= 74	Good
Medium risk	50 >= t <=59	Average

where't' is the overall students' achievement which serves as the target output in the network model

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Figures and Tables

It can be seen that, the enhanced network model records a very low error. Also Figure 4 is the target for the trained input attributes which comprises of 1300 datasets. Comparing this figure to the simulated results in Figures 5 and 6, it can be seen that Figure 6 which is the result of a simulation using the enhanced network shares similar characteristics with the target represented in Figure 4.

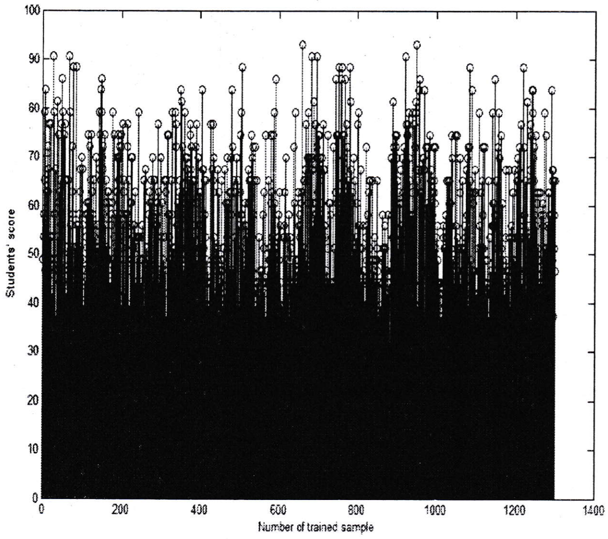

Fig. 4 The target for the trained inputs

It can be inferred from Figure 5 that the existing network did not have its trained dataset spanned through the input space enough, as only few similarities can be identified and several students are shown to have achieved almost the same score and above 90.

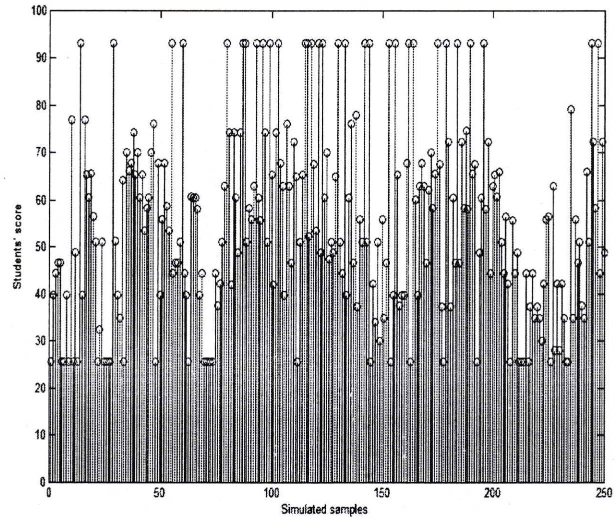

Fig. 5 Simulated results using the regular feed- forward network

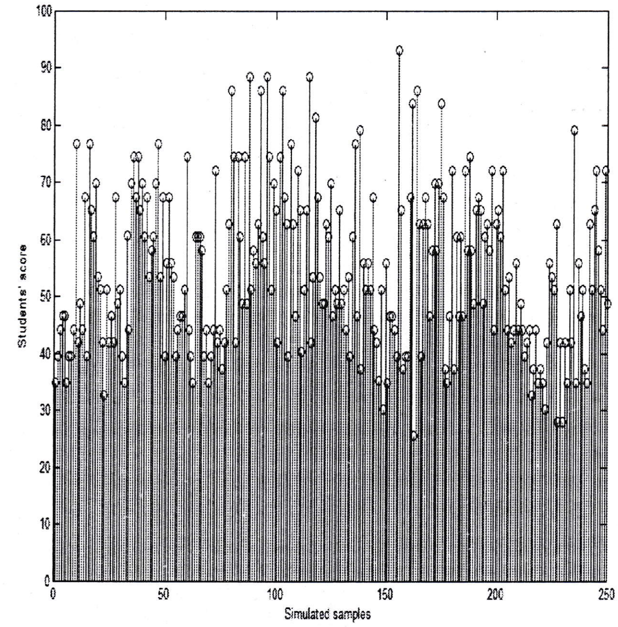

Fig. 6: Simulated results using the enhanced feed-forward network

The experimental results of partitioning dataset to 60%, 68%, 74% and 80% for training, while the remaining dataset takes equal percentages for validation and testing purposes is illustrated in Table 3. The enhanced network delivers the lowest error (see Table 3); the present study have therefore shown that, the enhanced network generalizes well than the existing network in its regular structure that has a default partition of 60%

TABLE III
MEAN ABSOLUTE ERRORS FROM DIFFERENT PARTITIONS

Training set (%)	Validation set (%)	Testing set (%)	Error
60	20	20	0.233208
68	16	16	0.023809
74	13	13	0.015179
80	10	10	0.615503

VI. CONCLUSION

This paper show the technique through which the performance of feed-forward neural network can be enhanced in order to boost the prediction accuracy of a network model. The configuration of the network and how the data to be trained are partitioned is paramount to the overall network performance. This is often the case whenever a model is to be constructed with varied data points. The results of comparing the two network models developed in the course of this study have shown that, the proposed approach is capable of producing a more accurate and reliable network model than the existing technique. While the proposed approach will be more useful when big or moderate data is to be trained, the network model created using the existing technique that train 60% of the data set can do well with small datasets. The study explores several options by which dataset can be partitioned to improve the performance of the network and a stable network model that generalizes well, especially, when new untrained inputs are to be predicted is achieved. The enhanced network model created in this study delivers consistent and relatively low error when evaluated using simulation methods. The network model created using the enhanced feed-forward technique is, therefore, suitable for creating a network model that can consistently predict well.

REFERENCES

[1] Suh, S.C., Practical Applications of DATA MINING. 2012: Jones & Bartlett Learning.

[2] Jiawei, H., K. Micheline, and P. Jian, DATA MINING Concepts and Techniques. 3rd ed. 2012: Morgan Kadfumann, Elsevier, USA.

[3] Erkaymaz, O., M. ÖZer, and N. YumuŞAk, 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, 2014. 22: p. 708-718.

[4] Negnevitsky, M., Artificial Intelligence A Guide to Intelligent Systems. 2011: Pearson Education Limited.

[5] Witten, I.H., E. Frank, and M.A. Hall, DATA MINING Practical Machine Learning Tools and Techniques. 3rd Edition ed. 2011: Morgan Kaufmann.

[6] Beale, M.H., M.T. Hagan, and H.B. Demuth, Neural Network Toolbox User's Guide. 2014a: The MathWorks, Inc.

[7] Rumelhart, D.E., G.E. Hinton, and R.J. Williams, Learning representations by back-propagating errors. Cognitive modeling, 1988.

[8] Haykin, S., Neural Networks and Learning Machines. 2009: Pearson Prentice Hall, USA.

[9] Mitsa, T., Temporal Data Mining. 2010, USA: CRC Press, Taylor & Francis Group.

[10] Cohn, D., L. Atlas, and R. Ladner, Improving generalization with active learning. Machine learning, 1994. 15(2): p. 201-221.

[11] Stanley, K. and R. Miikkulainen, Evolving neural networks through augmenting topologies. Evolutionary computation, 2002. 10(2): p. 99-127.

[12] Ampazis, N. and S.J. Perantonis, Two highly efficient second-order algorithms for training feedforward networks. Neural Networks, IEEE Transactions on, 2002. 13(5): p. 1064-1074.

[13] Thimm, G. and E. Fiesler. Evaluating prunning methods. in International Symposium on Artificial Neural Networks. 1995. Taiwan, Republic of China.

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

[15] Simard, D., L. Nadeau, and H. Kröger, Fastest learning in small-world neural networks. Physics Letters A, 2005. 336(1): p. 8-15.

[16] Issanchou, S. and J.P. Gauchi, Computer-aided optimal designs for improving neural network generalization. Neural Netw, 2008. 21(7): p. 945-50.

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

[18] Han, F., H.-F. Yao, and Q.-H. Ling, An improved extreme learning machine based on particle swarm optimization, in Bio-inspired computing and applications. 2012, Springer. p. 699-704.

[19] Zhao, G., et al. On improving the conditioning of extreme learning machine: a linear case. in Information, Communications and Signal Processing. ICICS 2009. IEEE.

[20] Thammasiri, D., et al., A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. Expert Systems with Applications, 2014. 41(2): p. 321-330.

[21] Kantardzic, M., DATA MINING: Concepts, Models, Methods, and Algorithms. 2nd ed. 2011: John Wiley & Sons Inc.

[22] Hall, M.A. and G. Holmes, Benchmarking attribute selection techniques for discrete class data mining. Knowledge and Data Engineering, IEEE Transactions on, 2003. 15(6): p. 1437-1447.