EduNet: A Deep Neural Network Approach for Predicting CGPA of Undergraduate Students

Md Rifatul Islam Rifat*, Abdullah Al Imran†, A. S. M. Badrudduza‡
*,‡Dept. of Electronics & Telecommunication Engineering
Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
†Dept. of Computer Science & Engineering
American International University-Bangladesh, Dhaka, Bangladesh
Email: {*irifat.ruet, †abdalimran, ‡asmb.kanon}@gmail.com

***Abstract*—Educational Data Mining (EDM) is an emerging research field concerned with the application of data mining, machine learning, and statistics in the discipline of education. Many researchers have already focused on EDM and exploring the educational data using several traditional data mining techniques to improve the educational performance of the students by extracting the concealed patterns and predicting the final outcome. In this study, we aim to propose a Deep Neural Network (DNN) based model to predict the final CGPA of the undergraduate business students with a minimal error than the traditional approaches. We have considered the performance of a decision tree model as the baseline performance. Experiments in this study have shown that our proposed DNN model can predict the CGPA with a significantly minimal error rate. To measure the performance of our model we have considered the three evaluation metrics namely Mean Squared Error (=0.008), Mean Absolute Error (=0.067), and Mean Absolute Percentage Error (=2.074). Our proposed model has successfully shown a promising prediction performance by reducing the MSE, MAE, and MAPE by 0.0146, 0.0431, and 6.043 respectively, compared to the baseline model.**

***Index Terms*—Deep Neural Networks, Regression, Educational Data Mining, CGPA Prediction.**

I. Introduction

In the modern era, education is the most important factor to change the world and thus many researchers have emphasized on examining the educational data and attempting to promote the quality of the education as well as improving the performance of the students. They have also been trying to create an ideal educational system [1]. Moreover, most of the university has already taken this as a challenge.

Since an educational system generates large-scale of data within a short period of time, the researchers have focused on the data mining and computational techniques to analyze the educational data and find insights from it. Fundamentally, data mining is an information extraction activity that identifies the concealed patterns in the data [2]. Nowadays, each institution is aiming to produce higher and exemplary education rates by employing various data mining methods.

The increasing emphasis of the researchers on data mining and education system has already originated a new research field named Educational Data Mining (EDM). It is an emerging research domain concerned with diverse type of analytical methods for investigating the educational data and use of those methods to provide a better understanding of the students [3]. Numerous data mining techniques are engaged in EDM such as decision trees, k-nearest neighbors, association rule, neural networks, genetic algorithms, exploratory factor analysis, and stepwise regression.

In most of the Asian countries, many vicissitudes are observable in the performance of the students during their undergraduate period. Thus, it creates a big problem for both the teachers and students to track the educational performance and take necessary steps to avoid the educational failures. So, an early prediction of the final CGPA can solve the problem and help the teachers, course-advisors, and instructors to identify the students at risk and also give high confidence to students in their studies.

In this study, we have proposed a Deep Neural Network (DNN) model to predict the the final CGPA of the undergraduate business students with a minimal error rate based on the transcripts data of first four semesters. We have considered the performance of a decision tree model as the baseline performance. Our experimental results have shown that the proposed DNN model can predict the final CGPA with Mean Squared Error (=0.008), Mean Absolute Error (=0.067), and Mean Absolute Percentage Error (=2.074). However, the comparison between the proposed model and the baseline model have concluded that our proposed DNN model has successfully performed a promising prediction by reducing the MSE, MAE, and MAPE by 0.0146, 0.0431, and 6.043 respectively.

II. Related Work

Since 1990s, numerous studies have been carried out in the area of educational improvement using computational technologies. In the current decade, a considerable amount of studies in this area have engaged with the use of different types of data mining approaches. Some researchers have employed a diverse type of machine learning algorithms to perform a different type of explorations such as predicting the performance of the students, selecting the most impactful predictors for more accurate predictions, and discovering the hidden knowledge from the relevant data. Now, we will discuss some relevant previous research in brief.

978-1-7281-3445-1/19/$31.00 ©2019 IEEE

To predict students' last semester GPA of the undergraduate period based only on the GPA of the preceding semesters the authors in [4] have used decision tree algorithm. They have also identified some impactful relations between the courses and final GPA. On the other hand, Sumitha et al. [5] have tried to propose a framework for exploring and classifying students' performance depending on academic data. According to the framework, first, they have identified the appropriate features by applying a ranking algorithm. And, then, they have employed a clustering approach named 'k-mean clustering'. They have also applied five efficient classification algorithms such as J48, Naive Bayes, SMO, Multilayer perceptron, and REP tree to predict the performance of the students and J48 is the one that outperforms the others with the highest accuracy (=97%).

Selecting the optimized features significantly enhance the learning performance of a machine learning model. Mueen et al. [6] have identified the appropriate attributes by applying the ranking algorithm. To properly train the models with the training data, they have applied the data partitioning method named 10-fold cross-validation. Three classifiers namely C4.5, Naive Bayes, and Multilayer Perceptron are tested on all of the 38 features to predict and explore students' educational performance. In this case, Naive Bayes has yielded the best accuracy (=86%).

Using the filter methods as well as the wrapper methods the prominent authors have conducted an experiment to propose an impactful feature subset [7]. The proposed feature subset contains 'attendance', 'previous failure', 'internal grades', 'internet usage hours', 'family relation quality', and 'health condition'.

Prediction of the ability of a student to complete his degree depending on his previous activity undoubtedly help the student to be careful. By considering this, the authors Daud et al. [8] performed this task with the application of generative and discriminative classification models. Then, they selected the best predictors with gain ratio and predicted the academic state of the students. They have applied three discriminative models (SVM, CART, and C4.5) and two generative models (Naive Bayes and Bayes Network). Among those SVM performs the best with f1-score 0.867.

Like the aforementioned study [5], the authors in [9] have provided a generalized framework for both the prediction and pattern analysis. Then, they have validated their proposed framework by conducting an extensive experiment on real data. In their experiment, they have used several state-of-the-art algorithms which are GBT, Tree Ensemble, SVM, Decision Tree, Random Forest, and KNN to predict the academic performance. Random forest results in the best accuracy of 94.1%. For pattern analysis, they have extracted a tree using the decision tree algorithm and found 'Computer in Business' as the most impactful course.

A discriminative analysis has been conducted between Multilayer Perceptron and GRNN to determine the model that performs the best in predicting the educational performance of the students depending only on previous results [10]. The authors performed a train-validation-test split with a ratio of 60:20:20, where 60% used for training the model, 20% for validation, and other 20% for testing the model's performance. The performance has been evaluated with the evaluation metrics: mean squared error, accuracy, and ROC. According to the comparison, they concluded that GRNN yields better accuracy (=95%) than Multilayer Perceptron.

III. Problem Background

If we focus on Asian countries, we can observe that there exists a non-linear relationship between the grades, semester GPAs, and overall educational performance of an undergraduate student. If we segregate all of the students into three categories, we can see that one category perform better in the beginning but worse in the ending of the undergraduate period while the second category shows the reverse scenario. And, a few students, the third category, always perform either better or worse. These different types of cases raise various difficulties for the teachers or instructors to detect the students who are at risk. Predicting the final CGPA of the students, undoubtedly, reduce the difficulties for the teachers or instructors to easily identify and guide the students who are at risk.

At the central point of the undergraduate period, some students are attacked by the frustration of failure and this frustration significantly influence their performance that increases the possibility of falling out. So, the early prediction of the final CGPA undoubtedly can help the teachers or instructors to take a proper decision for the improvement of students' performance. Moreover, it'll also carry away the frustration from the affected students and reinforce their mind to perform better.

From the researchers' perspective, the rapid ups and downs in the educational performance of students throw a challenge to the researchers to predict the final CGPA. To handle the aforementioned problems, we have aimed to predict the final CGPA using a DNN approach only by utilizing the students' transcript data of the first four semesters.

IV. Description of Dataset

In this research, we have prepared a real dataset by collecting the transcript data from the result sheet of the marketing department of a reputable public university in Bangladesh. This dataset composed of the transcript data of the students whose passing year is 2013, 2014, 2015, and 2016. The collected raw data, that have contained 398 instances in total, have been manually concocted and arranged in an excel spreadsheet. The features that are contained in the processed dataset are students name and ID, gender, Semester GPA, course grade, state of the students depending on the GPA of the relevant semester, and final CGPA. Each of the courses is defined by a distinct code that are 101, 102, 103, 104, 105, 111,112, 113, 114, 115, 201, 202, 203, 204, 205, 211, 212, 213, 214, 215, 301, 302, 303, 304, 305, 311, 312, 313, 314, 315, 401, 402, 403, 404, 405, 411, 412, 413, 414, 415 and detail are available in [11]. However, we have used the courses that are only included in the first four semesters to predict the

final CGPA. The courses that are included in the first four semesters ranges from 101 to 215.

V. Data Preprocessing

To solve a real-world problem using machine learning, it is indispensable to preprocess the raw data properly for achieving a better result from the applied model as it directly affects the ability of the model to learn. Real-world data are generally incomplete, noisy, corrupted, inconsistent, and imbalanced and so it is essential to preprocess the data before feeding it into the model. Data preprocessing is an integral step such as data cleaning, data integration, feature encoding, instance selection, normalization, transformation, feature extraction, and selection. In this study, we have structured this phase according to the form of our dataset which includes missing value handling, feature engineering, feature scaling, feature encoding, data partitioning. All the tasks are discussed briefly in the subsections below.

A. Feature Engineering

Feature Engineering is the method of extracting new characteristics by utilizing the current features into formats that are suitable for the machine learning model [12]. We have performed feature engineering techniques to extract the gender of the students based on their name. Also, we have developed distinct feature columns that indicate the state of the students after finishing each of the semesters based on their acquired GPA. According to Table 1, we have employed four distinct terminologies namely '*Excellent*', '*Very Good*', '*Good*', and '*At Risk*' to indicate the state of the students.

TABLE I
GPA range for every defined state

GPA	Terminology
3.5 or above	Excellent
3.00 to less than 3.50	Very Good
2.50 to less than 3.00	Good
Less than 2.5	At Risk

B. Feature Encoding

Feature encoding is the technique that represents the categorical data into the numeric form. The neural networks can only learn from the numeric data that is why it is mandatory to encode the categorical data into numbers before feeding the data to the model. In this study, the categorical features that include in our dataset are the gender and the state of the students mentioned above. In order to encode these variables into a compatible numeric form, we have applied the encoding technique named One-hot encoding. This technique generates a binary column for each of the n categories of a categorical variable, where n is the number of unique values contained in that variable [13].

C. Feature Scaling

Feature scaling [15] is a crucial step in the data preprocessing pipeline that is used for data standardization. This includes a few techniques that scale the data into a fixed range to bring all the features to the same level of magnitudes, which suppresses the effect of outliers. It is essential for training a neural network based model as it helps the weights to converge more quickly. For feature scaling, we have applied min-max scaling to each feature column, where the new value X_{sc} of a sample X can be calculated as follows:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Here, X_{min} is the smallest value in a feature column, and X_{max} is the largest value, respectively. This scaling brings the new value between 0 and 1.

D. Data Partitioning

To properly access the whole dataset, we have applied the training-validation-testing approach in our study. In a neural network model, it is a basic need to tune the hyperparameters and be cautious about the model overfitting and underfitting. A validation set is crucial for properly tuning the hyperparameters and optimize the best model. From the total number of 372 instances, we have used 68% (=261) instances for training, 20% (=66) for validation, and 12% (=45) for testing the model.

VI. Methodologies

In this study, we have maintained a proper workflow for conducting our experiment [14]. The entire workflow basically contains three consecutive steps such as designing the architecture of the DNN model, selecting the evaluation metrics, and defining a baseline performance score for the model. All the implementations of these three steps have been performed using the Python programing language.

A. The Architecture of the DNN Model

DNN models are basically comprised of three major layers: the input layer, the hidden layers, and the output layer. The domain-specific architecture of this model can be defined by some parameters such as the number of nodes or neurons in each layer, the number of hidden layers, the activation function, the optimization algorithm, the learning rate of the network, the number of epochs, batch size, and so on. All of these parameters need to be involved during the design of a network and these parameters lead the learning process and the end performance of the model. Thus, it is indisputably a challenging task to select the value of these parameters . Fig. 1 represents the visual architecture of our proposed model.

1) The Number of Neurons in the Input and Output Layers: A neural network architecture begins with configuring the hyperparameters which are the number of neurons in the input and output layer. The value of these hyperparameters depends on the shape of the training data. Our training data has a shape of (261, 43), which means there are 261 instances and 43 attributes. We have specified 43 neurons in the input layer as

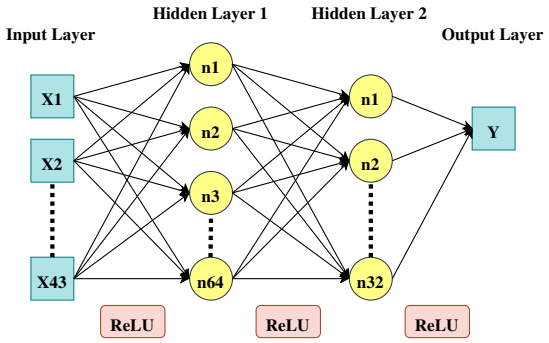

Fig. 1. DNN Architecture.

there are 43 input features in the training data. In contrast, we have specified only one neuron for the output layer as the problem is a regression problem and our expected outcome is a single numeric value.

2) The Number of Hidden Layers and Neurons: Determining the number of hidden layers and the number of neurons in each hidden layer is a crucial task as it characterizes the overall neural network architecture. A study conducted by Jeff Heaton [16] has concluded that networks with a single hidden layer can approximate any function which carries a consecutive mapping from one finite space to another. In contrast, networks with two hidden layers can describe an arbitrary decision boundary to arbitrary accuracy with rational activation functions. And, networks with more than three hidden layers can learn complicated designs and deliver automated feature engineering. As our training data is not very large, more than two hidden layers will make our model very complex, computationally inefficient, and may cause overfitting to the training data. So, we have designed our neural network architecture employing 2 hidden layers.

Again, to select the number of neurons in the hidden layer, we have applied a rule of thumb from [17]. The book has enunciated that the number of hidden neurons in the first hidden layer should be defined according to the number of input dimensions. If x be the final number of input dimension in a given training dataset, then we should use the nearest number to $2x$ in the power of 2. As our final input dimension is 43, we have employed 64 neurons for the first hidden layer which is nearest to $2x$ in the power of 2. Since the second hidden layer is the last hidden layer before the output, we have employed 32 neurons which the nearest number to $\frac{2x}{2}$ in the power of 2.

3) Activation Functions: Activation functions are an extremely important feature of the neural networks that perform functional mappings between the inputs and response variable. Basically, activation function is a non-linear transformation to the input signal making it capable to learn and perform more complex tasks. After computation, each hidden neuron of a hidden layer generates a single output which is then used as an input in the next layer in the stack. In this study, we have applied the activation function ReLU which is briefly discussed below.

Rectified Linear Unit (ReLU): ReLU [18] is the most widely used activation function while designing a DNN model. If the input is negative the function will convert it to zero and the neuron does not get activated, but for any positive value x, it returns that value back. It is defined as-

$$f(x) = max(0, x) \tag{2}$$

where x is the input to a neuron. So, this is the main advantage of using the ReLU function over other activation functions as it does not activate all the neurons at the same time. Activating a few neurons at a time making the network efficient and easy for computation.

4) Optimization Algorithm: In DNN, an optimization algorithm plays the most important role for obtaining a desirable result from the model. It shapes the neural network model into its most accurate possible form by adjusting the weights and train up model with an optimized cost function. Several optimization algorithms are used in neural networks such as Adam, Gradient Descent, Adagrad, RMSprop, and so on. However, for our experiment, we have applied the optimization algorithm Adam which is briefly described below.

Adaptive Moment Estimation (Adam): The Adam [19] optimization algorithm is an extension to stochastic gradient descent that updates network weights iterative based in training data and works well across a wide range of deep learning architectures. Indeed, this adaptive learning rate optimization algorithm is a combination of gradient descent with momentum and RMSprop algorithms. Adam has some advantages over other optimization algorithms such as it has relatively low memory requirements, invariant to diagonal rescale of the gradients, hyper-parameters have intuitive interpretation and typically require little tuning, and so on.

5) Evaluation Metrics: To evaluate the performance of our proposed model with measuring the error, we have applied three evaluation metrics namely Mean Squared Error, Mean Absolute Error, and Mean Absolute Percentage Error. All the metrics have been elected by considering their benefits and application. The details of these metrics have been discussed below.

Mean Squared Error (MSE): MSE is basically a measure of how close our fitted line is to data points. For each of the data points, it calculates the squared difference between the predicted value and the target value and then averages those differences. The smaller value of this calculation indicates the better fitting of the model. The following equation defines the mathematical formulation of MSE, that is:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

Mean Absolute Error (MAE): MAE is extensively used for forecasting accuracy where the error is calculated as an average of absolute differences between the predictions and the target values. Basically, it is a linear score which means that all the individual differences are weighted equally in the average. One of the important advantages of this metric is that

it penalizes huge errors that not as that badly as MSE does. Thus, it is not that sensitive to outliers as mean squared error. MAE can be formulated as:

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4}$$

Mean Absolute Percentage Error (MAPE): MAPE is a measure of the prediction accuracy that is used as a loss function for regression problems in machine learning. It is a non-scaled error metric and expressed as the relative error preference. For each data points, the absolute error is divided by the target value, yielding a relative error. The main advantage of this measure is its simplicity and it is easy to understand because it provides the error in terms of percentages. Thus, if the average range of the prediction is already known, then the predictions can be easily estimated. Mathematically, it can be calculated by the following equation:

$$\frac{100\%}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \tag{5}$$

6) Defining Baseline Performance: Defining the baseline performance is a crucial task to monitor and assess the effectiveness of the neural network model in the training phase. We have considered the performance of a Decision Tree [20] as the baseline for all of the test samples. We have conducted an experiment on our dataset by applying the Decision Tree and the baseline performance we have attained from the decision tree model are Mean Squared Error (=0.02390), Mean Absolute Error (=0.1114), Mean Absolute Percentage Error (=8.0047). To obtain a promising result, our model should offer a performance error less than the baseline error.

VII. Result and Analysis

In this study, we have measured the performance of our proposed neural network model using three evaluation metrics namely MSE, MAE, MAPE. Here, we have applied the MAE as the loss function and have trained our model up to 50 epochs. Fig. 2 presents the losses with respect to the epochs.

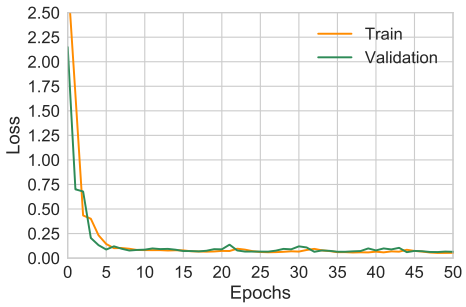

Fig. 2. Train vs Validaion loss across epochs.

The above visualization shows the learning rates of the model over the epochs. It can be apparent that the training and validation losses have been converged at 10^{th} epochs. Here, the validation loss has been quite compatible in converging with training loss indicates a well-fitted model.

After the training and validation phase, we have applied the model on the test data to predict the final CGPA. The test data set have contained 45 instances. Table 2 presents the training, validation, and test result obtained from our model.

TABLE II
Performance of the Proposed Model

	MSE	MAE	MAPE
Training	0.13	0.119	3.536
Validation	0.071	0.116	3.517
Test	0.008	0.067	2.074

From Table 2 it can be observed that the validation result is very close to the training result, which indicates that our proposed model has learned well from the data without overfitting. Interestingly, in all the phases the results are quite compatible with very minimal variations.

In order to grasp the variations in actual and predicted CGPA more expressly, we have visualized all the 45 test cases into a line graph in the following Fig. 3.

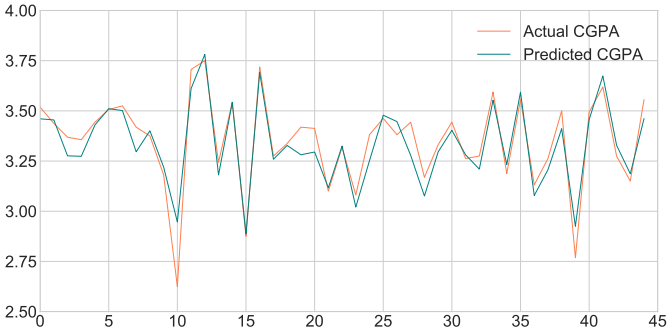

Fig. 3. Actual CGPA vs Predicted CGPA.

From the line graph in Fig. 3 it can be clearly observed that except few points, all the predicted values have a trivial variation with the actual values. For most of the samples, the model have fitted very well and predict the actual value with a minimal error. From all of the above analysis and error metrics we can conclude that our model has learned the underlying pattern from the educational data very well and can promisingly predict the actual CGPA of the undergraduate students.

The following Table 3 shows the comparison between the performance of the baseline model and the proposed model.

TABLE III
Comparison between the Baseline and the Proposed Model

	Baseline Performance (Decision Tree)	Proposed Model Performance (Deep Neural Network)
MSE	0.0226	0.008
MAE	0.1101	0.067
MAPE	8.1168	2.074

VIII. Conclusion

In this study, we have proposed a DNN based model in order to predict the final CGPA of the undergraduate business students with an error rate smaller than the traditional models. The dataset we have used was a real-world dataset collected from a reputed university in Bangladesh. We have performed a bunch of data preprocessing tasks such as feature engineering, scaling, encoding, and data partitioning to prepare our dataset in a suitable form for training the model. In the DNN architecture, we have selected the hidden layers, neurons, optimization algorithm and have tuned the hyperparameters by manual search and practical judgment. The performance of the model has been evaluated by using three evaluation metrics namely MSE, MAE, and MAPE. To examine the training and validation losses, we have used MAE as the loss metric. The training and validation performance of our proposed model indicates that the model has consistently learned from both training and validation data with very less deviation. In the test phase, our proposed model has also yielded very promising result such as 0.008 for MSE, 0.067 for MAE, and 2.074 for MAPE. And, our proposed model have significantly reduced the errors MSE, MAE, and MAPE by 0.0146, 0.0431, and 6.043 respectively than the baseline decision tree model. Most of the predicted points are very close to the true values and our proposed model performance have proved its consistent learning from the transcript data and showed a promising prediction performance.

In our future study, we aim to propose a framework that will be efficiently applicable for the prediction from this type of information. Furthermore, we will also incorporate more attributes such as age, study behavior, study time, family support, and disability for further analysis of the student performance.

References

[1] Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. Expert systems with applications, 41(4), 1432-1462.

[2] David, J. (2007). Hand. Principles of data mining. Drug safety, 30(7), 621-622.

[3] Sukhija, K., Jindal, M., & Aggarwal, N. (2016). Educational data mining towards knowledge engineering: a review state. International Journal of Management in Education, 10(1), 65-76.

[4] Al-Barrak, M. A., & Al-Razgan, M. (2016). Predicting students final GPA using decision trees: a case study. International Journal of Information and Education Technology, 6(7), 528.

[5] Sumitha, R., Vinothkumar, E. S., & Scholar, P. (2016). Prediction of Students Outcome Using Data Mining Techniques. International Journal of Scientific Engineering and Applied Sciences, 2(6).

[6] Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and predicting students' academic performance using data mining techniques. International Journal of Modern Education and Computer Science, 8(11), 36.

[7] Chaudhury, P., & Tripathy, H. K. (2017). An empirical study on attribute selection of student performance prediction model. International Journal of Learning Technology, 12(3), 241-252.

[8] Daud, A., Aljohani, N. R., Abbasi, R. A., Lytras, M. D., Abbas, F., & Alowibdi, J. S. (2017, April). Predicting student performance using advanced learning analytics. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 415-421). International World Wide Web Conferences Steering Committee.

[9] Rifat, M. R. I., Al Imran, A., & Badrudduza, A. S. M. (2019). Educational Performance Analytics of Undergraduate Business Students, International Journal of Modern Education and Computer Science(IJMECS), Vol.11, No.7, pp. 44-53, 2019.DOI: 10.5815/ijmecs.2019.07.05

[10] Iyanda, A. R., Ninan, O. D., Ajayi, A. O., & Anyabolu, O. G. (2018). Predicting Student Academic Performance in Computer Science Courses: A Comparison of Neural Network Models. International Journal of Modern Education and Computer Science, 10(6), 1.

[11] Department of Marketing. Website: http://dept.ru.ac.bd/marketing/index.php/programs/bba (Accessed on: 05 February 2019).

[12] Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. " O'Reilly Media, Inc.".

[13] Marko. Mooi Sarstedt (Erik.). (2018). Concise Guide To Market Research: The Process, Data, And Methods Using Ibm Spss Statistics. Springer-verlag Berlin An.

[14] Al Imran, A., Amin, M. N., Rifat, M. R. I. & Mehreen, S. (2019). Deep Neural Network Approach for Predicting the Productivity of Garment Employees. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE.

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[16] Heaton, J. (2008). Introduction to neural networks with Java. Heaton Research, Inc.

[17] Moolayil, J. (2019). Learn Keras for Deep Neural Networks. Apress. https://doi.org/10.1007/978-1-4842-4240-7

[18] Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Erratum: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789), 947–951. https://doi.org/10.1038/35016072

[19] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[20] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.