A Novel Hybrid Approach Of Adaboostm2 Algorithm And Differential Evolution For Prediction Of Student Performance

Samuel-Soma M. Ajibade, Nor Bahiah Binti Ahmad, Siti Mariyam Shamsuddin

Abstract: The prediction of performance of student is a very important task for institutions of higher learning. The academic performance of students aids the teachers, instructors and management of institutions to identify low performing students and then more attention is given to them so as to enhance their performances. In previous studies, various elements and methods have been applied to identify and enhance the performances of students. In this work, we made use of data mining classifcation techniques to improve the prediction accuracy of student performance. In recent times, classification accuracy has been enhanced thru the use of ensemble techniques and combining multiple classifiers. In this paper, we have made use of an efficient AdaBoost ensemble technique called AdaBoostM2 and we combined it with a metaheuristic optimization algorithm known as Differential Evolution (DE) to produce a novel algorithm called "ADDE". This new algorithm is implemented on the KalBoard 360 educational dataset and the results displays that is very efficient in reducing the weak learners and thereby increases the prediction accuracy. The new algorithm has therefore shown better result in reducing the computation complexity.

Index Terms: AdaBoost, AdaBoostM2, Classification techniques, Data mining, Differential Evolution, prediction model, Student Performance

1 INTRODUCTION

The student performance prediction is an important concern in e-learning environs. The performance of student depends on different factors, for example, individual, social, natural and psychological factors. Data Mining techniques are capable devices used to achieve these goals. These methods are utilized to find hidden designs and relations on an extensive measure of data that might be useful in making of decisions. Classification is a standout amongst the most helpful data mining methods used for prediction with e-learning, it maps information into pre-defined clusters of classes, and it is frequently called supervised learning. Predictive models plan to predict the obscure estimations of factors of interest given known estimations of different factors. The high accuracy of performance prediction of students is highly advantageous for detecting students that are performing low. It is necessitated that the detected students can be helped more by the instructor or tutor in order to enhance their performance [1] [9]. The main objective of this research is to build a model for prediction thru analyzing the features that has effect on the performance of students by the use of the Education dataset gotten from Learning Management System (LMS) known as 'Kalboard 360'. The data was collected using a learner activity tracker tool known as Experience API (XAPI). In past work [2], feature selection methods were applied in order to choose the most important features so as to minimize the dataset size, after concluding the experiment, the outcomes demonstrate that every one of the classifiers applied gives the best performance with 3-7 features out of 16 [2]. The prediction model 'ADDE' is proposed and it is actualized by utilizing the AdaBoost algorithm with differential evolution (DE), and it is thereby discovered that our proposed algorithm decreases the computation complexity is model whereas keeping up high detection accuracy when compared to other models. Therefore, this model will predict whether the student will answer false or true for specific problems.

2 LITERATURE REVIEW

The investigation in [3] proposed an enhanced decision tree to identify the pointers of student dropouts. The research gathers the dataset of 240 students through a study and applies Correlation based Feature Selection (CFS) algorithm (Filter feature selection algorithm) in pre-processing step. The classification precision of the model displays over 90%. Be that as it may, the investigation made use of only one dataset. The analysis in [4] assessed six feature selection algorithms to predict the academic outcome of some higher secondary students. The after effect of the research concludes that Voted Perceptron, and One Rule (OneR) demonstrates high predictive performance with the element subsets increased via feature selection algorithms. Moreover, Information Gain (IG) and CFS demonstrates better ROC value and F-measure values on the dataset of the higher secondary school. n investigation to predict the academic outcome of student in secondary school at Tuzla was discussed in [5]. The research utilized Gain Ratio (GR) feature selection algorithm on the dataset with 19 features. The outcome gotten by applying Random Forest classification (RF) algorithm demonstrates the best results as regarding the accuracy of prediction. Previous research carried out by Amrieh utilizing Naive Bayes, Decision Tree, and ANN algorithms [6]. The embedded method as the feature selection technique using Bagging, Boosting, and Random Forest. From all the results, ANN has the highest accuracy is ANN utilizing Boosting feature selection. Accuracy using behavioral features can increase as much as 22.1%, while after using feature selection, the accuracy increases to 25.8%. [7] evaluated how well student accomplishments can predict graduate-level performance. The information of 171 student records in the Bachelor and Master program in Computer Science at ETH Zurich, Switzerland was utilized. Employing linear regression models in blend with different variable-selection techniques, their discoveries demonstrates that undergraduate level performance can demonstrate as much as 54% of the variance in graduate-level performance. The year three (3) grade point average was distinguished as the most important informative variable, whose impact surpasses the one of grades earned in challenging year one (1) [7].

3 THE PROPOSED PREDICTION MODEL 'ADDE'

3.1 Boosting

Boosting boosts the performance of the weak classifier to a strong level. It generates sequential learning classifiers using resampling (reweighting) the data instances. Initially equal uniform weights are assigned to all the instances. During each learning phase a new hypothesis is learned and the instances are reweighted such that correctly classified instance having lower weight and system can concentrate on instances that have not been correctly classified during this phase having higher weights. It selects the wrongly classified instance, so that they can be classified correctly during the next learning step. This process continuous tills the last classifier construction. Finally, the results of all the classifiers are combined using majority voting to find the final prediction. AdaBoost [8] is a more general version of the Boosting algorithm.

3.1.1 Boosting Algorithm: AdaBoost

In this subdivision, we describe our boosting algorithm which is known as AdaBoost. [8] proposed AdaBoost and defines two versions of Adaboost and they labelled it AdaBoost.M1 and AdaBoost.M2. These two versions are equivalent for binary classification problems and are different just in their handling. In previous study, [10] made use of the AdaBoost.M1 to propose a prediction model. The main disadvantage of AdaBoost.M1 is that it is unable to handle weak hypotheses with error greater than 1/2. The expected error of a hypothesis which randomly guesses the label is 1-1/2, where k is the number of possible labels. Thus AdaBoost.M1 requirement for $k = 2$ is that the prediction is just slightly better than random guessing. However, when $k > 2$, the requirement of AdaBoost.M1 is stronger than that, and this requirement will be difficult to fulfil. However, the second version of AdaBoost which is known as "AdaBoost.M2" tries to overcome these challenges by spreading out the communication between the boosting algorithm and the weak learner. Firstly, the weak learner is allowed to create more expressive hypotheses whose output is a vector in $[0,1]^k$, rather than a single label in Y. Intuitively, the yth component of this vector represents a "degree of belief" that the correct label is y. The components with values close to 1 or 0 correspond to those labels considered to be plausible or implausible, respectively [8]. In our research work, we therefore made use of "AdaBoost.M2" for classification problems. Figure 1 shows the Adaboost Algorithm for binary classification problems. Although the weak learning algorithm has more expressive power, a more complex requirement is positioned on the performance of the weak hypotheses. Instead of making use of the normal prediction error, it is required that the weak hypotheses perform better with respect to a more refined error measure which is known as the pseudo-loss. Not like ordinary error that is computed with respect to a distribution over examples, pseudo-loss is computed with respect to a distribution over the set of all pairs of examples and incorrect labels. Through manipulating this distribution, the boosting algorithm can focus on the weak learner not only on hard-to-classify examples, but more specifically, on the incorrect labels that are hardest to discriminate. We will see that the boosting algorithm AdaBoost.M2, which is based on these ideas, achieves boosting if each weak hypothesis has pseudo-loss slightly better than random guessing

Input: sequence of m examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ with labels $y_i \in Y = \{1, \ldots, k\}$
weak learning algorithm **WeakLearn**
integer T specifying number of iterations

Let $B = \{(i, y) : i \in \{1, \ldots, m\}, y \neq y_i\}$
Initialize $D_1(i, y) = 1/|B|$ for $(i, y) \in B$.
Do for $t = 1, 2, \ldots, T$

1. Call **WeakLearn**, providing it with mislabel distribution D_t.
2. Get back a hypothesis $h_t : X \times Y \to [0, 1]$.
3. Calculate the pseudo-loss of h_t: $\epsilon_t = \frac{1}{2} \sum_{(i,y) \in B} D_t(i, y)(1 - h_t(x_i, y_i) + h_t(x_i, y))$.
4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.
5. Update D_t: $D_{t+1}(i, y) = \frac{D_t(i, y)}{Z_t} \cdot \beta_t^{(1/2)(1 + h_t(x_i, y_i) - h_t(x_i, y))}$
   where Z_t is a normalization constant (chosen so that D_{t+1} will be a distribution).

Output the hypothesis: $h_{fin}(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t} \right) h_t(x, y)$.

Fig. 1. *The AdaBoostM2 Algorithm.*

3.2 Overview of Differential Evolution

DE is an evolutionary algorithm introduced by [13] for global optimization problems over continuous spaces [13]. It is easy and requires only few control parameters. Because of these reasons, DE has been preferred by researchers in many fields. DE starts with the randomly generated initial values for individuals (i.e., solution) in the search space, then, mutation and crossover operators, and selection process are applied to individuals to generate a new population. The steps of DE are as follows:

Step 1. Initialization: Generally, the initial population is generated by using the pre-determined minimum and maximum bounds which are defined as in (1).

$$X^j_i = X^j_{min} + \text{rand}(0,1).(X^j_{max} - X^j_{min}) \quad (1)$$

Where j=1,2,…,D, X^j_{max} and X^j_{min} are the maximum and minimum bounds of the j th parameter of the problem while rand (0,1) is a random value between 0 and 1.

Step 2. Fitness evaluations of the individuals in the population: In this step, fitness value is calculated for each individual in the population.
Step 3. Mutation process: DE employs the mutation operation to produce a mutant vector V_i with respect to *j* th individual in the population, and Xi so-called source vector (i.e., individual), in the current population. The most frequently used mutation operator in the DE implementations is given in (2).

$$V^j_i = X^j_{r1} + F.(X^j_{r2} - X^j_{r3}) \quad (2)$$

where F is the scaling/mutation factor having values in the range of [0,2], Xr1, Xr2, Xr3 are source vectors which are randomly chosen from the population, and r1, r2, r3 and *i* must be different from each other (r1 ≠ r2 ≠ r3 ≠ i).

Step 4. Recombination (crossover) process: After creating the mutant vector, crossover operation is applied to the source vector Xi and its corresponding mutant vector Vi to generate a trial vector Ui. This process is performed by using (3).

$$U_i^j = \begin{cases} V_i^j, & if (rand[0,1) \leq CR \text{ or } j = j_{rand}) \\ X_i^j, & \text{otherwise} \end{cases} \quad (3)$$

where the crossover rate (CR) is a user-specified constant within the range of [0,1], which controls the rate of parameter values copied from the mutant vector. jrand is a randomly chosen integer in the range of [1, D] where D is the number of variables in the problem to be solved. Step 5. Fitness evaluation and selection: The fitness function value of the trial vector is compared with that of source vector. If the trial vector has higher fitness function value than the source vector, the trial vector replaces the source vector and it is included into the population of the next generation. Otherwise, the source vector remains unchanged in the population for the next generation. Steps 3, 4, and 5 are repeated for each individual until a predetermined termination criterion is met. The best solution in the population is the (sub)optimum solution.

3.3 Overview of Proposed Model 'ADDE'

AdaBoost algorithm, when compared likened to most other learning algorithms, is less susceptible to difficulty of overfitting than them and this is in light of the fact that boosting is very sensitive to noisy data and outliers [12]. Therefore, the difficulty of overfitting can be caused as a reason of mislabeled cases. The new classifiers pay more attention on wrongly classified cases, which as a result gives a huge amount of classifiers that are weak in order to attain a better performance [11]. In our research, a new boosting algorithm has been developed which is known as "ADDE". This new algorithm optimizes the amount of weak classifiers and their respective weights by utilizing a differential evolution (DE) algorithm, so as to enhance the boosting performance. The DE is able to regulate the way outliers are affected, thereby selecting a suitable fitness function which confines that number of classifiers that are weak and hence enhances the accuracy of prediction. This new model has more benefits than other boosting algorithms in that it decreases the complexity of the model. The ADDE algorithm is shown in Figure 2

Input: a set S of m instances: $S=\{(x_1,y_1),\dots,(x_m,y_m)\}$ with labels $y_i \in Y = \{1,\dots,k\}$, Input population size (N), Problem size/dimension (D), Weighting factor (F), Crossover rate (CR), Number of Iterations (T).

Initialize: a randomly generated vector population by the use of the maximum and minimum bounds as defined in Eq. 1.

Evolve: t=1, 2,…, T.

Step 1: Generate a vector population X

Step 2: Fitness evaluation of each individual in the population

Step 3: Make use of the vector X to update the weak learners (T) and their respective weights

Step 4: Produce new generation of solutions by utilizing the DE operations (mutation, crossover and selection).

Step 5: If termination criteria is satisfied, stop and return the desired optimize solution; else loop (go back to step 2).

Fig. 2. *The proposed procedure of ADDE.*

4 RESULTS AND DISCUSSION

In this present study, the dataset lengthens into 500 students with 16 features. Table 6 displays the attributes/features, data description and data type of the dataset.

TABLE 1
PROPOSED ADDE CLASSIFIER WITH ADABOOSTM2 CLASSIFIER USING NAÏVE BAYES (NB)

Case No	**nWeakLearn AdaBoostM2**	**nWeakLearnADDE (Proposed)**	**No of Generation**	**Population size**	**AdaBoost Accuracy**	**ADDE Accuracy (Proposed)**
1	20	16	10	10	0.6944	0.7183
2	53	43	20	10	0.6825	0.7183
3	85	69	30	15	0.6944	0.7183
4	118	96	40	20	0.6944	0.7183
5	150	122	50	35	0.6825	0.7183
Average	**71**	**69**			**0.6896**	**0.7183**

The table 1 showed that the number of weak learners of the proposed classifier trained by AdaBoostM2 reduced by about 3% owing to the utilization of the DE optimization whereas the accuracy of the proposed algorithm increased by 4%.

TABLE 2

PROPOSED ADDE CLASSIFIER WITH ADABOOSTM2 CLASSIFIER USING DECISION TREE (DT)

Table 2 illustrated that the number of weak learners of the proposed classifier trained by AdaBoostM2 reduced by 78.9% owing to the utilization of the DE optimization whereas the accuracy of the proposed algorithm increased by 9.3%.

Case No	nWeakLearn AdaBoostM2	nWeakLearnADDE (Proposed)	No of Generation	Population size	AdaBoost Accuracy	ADDE Accuracy (Proposed)
1	20	16	10	10	0.6310	0.8075
2	53	43	20	10	0.7222	0.7659
3	85	4	30	15	0.7262	0.7738
4	118	5	40	20	0.7262	0.7738
5	150	6	50	35	0.7262	0.7738
Average	**71**	**15**			**0.7064**	**0.7790**

TABLE 3

PROPOSED ADDE CLASSIFIER WITH ADABOOSTM2 CLASSIFIER USING DISC

Table 3 displayed that the number of weak learners of the proposed classifier trained by AdaBoostM2 reduced by about 90.1% owing to the utilization of the DE optimization whereas the accuracy of the proposed algorithm increased by 4.5%.

Case No	nWeakLearn AdaBoostM2	nWeakLearnADDE (Proposed)	No of Generation	Population size	AdaBoost Accuracy	ADDE Accuracy (Proposed)
1	20	16	10	10	0.6746	0.7063
2	53	3	20	10	0.6746	0.7063
3	85	4	30	15	0.6746	0.7063
4	118	5	40	20	0.6746	0.7063
5	150	6	50	35	0.6746	0.7063
Average	**71**	**7**			**0.6746**	**0.7063**

TABLE 4

PROPOSED ADDE CLASSIFIER WITH ADABOOSTM2 CLASSIFIER USING KNN

Table 4 described that the number of weak learners of the proposed classifier trained by AdaBoostM2 reduced by about 62% owing to the utilization of the DE optimization whereas the accuracy of the proposed algorithm increased by 19.3%.

Case No	nWeakLearn AdaBoostM2	nWeakLearnADDE (Proposed)	No of Generation	Population size	AdaBoost Accuracy	ADDE Accuracy (Proposed)
1	20	2	10	10	0.6944	0.8671
2	53	3	20	10	0.6944	0.8611
3	85	4	30	15	0.6944	0.8770
4	118	5	40	20	0.6944	0.8413
5	150	122	50	35	0.6944	0.8540
Average	**71**	**27**			**0.6944**	**0.8601**

From the above analysis, the proposed ADDE classifier with AdaBoostM2 using KNN gave the highest accuracy of 86.01% in which the accuracy of the proposed algorithm increased by 19.3%. Furthermore, a comparison of proposed algorithm prediction accuracy with other classification algorithms that all made use of the Kalboard 360 educational dataset is shown in Table 6. It is seen in Table 6, that our proposed algorithm outshines others.

TABLE 5

COMPARISON OF SOME CLASSIFICATION ALGORITHM WITH OUR PROPOSED ALGORITHM

Authors	Classification Algorithms	Prediction Accuracy
Pandey, M., & Taruna, S. (2014)	Random Forest	75.9%
Amrieh, E. A. et. al., (2015)	ANN	73.8%
Stapel, M. et.al., (2016)	Hybrid of Decision Tree with Adaboost	74.0%
Rahman, M. H., & Islam, M. R. (2017)	Hybrid of Ensemble filtering with ANN	84.3%
Rahman, L. et. al., (2017)	ANN	79.4%
Adejo, O. W., & Connolly, T. (2018)	Hybrid Stacking ensemble technique	81.7%
Alam, M. M. et. al., (2018)	ANN	85.0%
Bendangnuksung & Prabu P. (2018)	DNN	84.3%
Proposed Model	**ADDE**	**86.01%**

TABLE 6
STUDENT FEATURES DESCRIPTION AND DATA TYPE

Feature Cate-Gory	Features	Description	Data Type
Demographical Features	Nationality	Student nationality	Nominal
	Gender	The gender of the student (female or male)	Nominal
	Place of Birth	Student's place of birth (Kuwait, Jordan, Lebanon, Saudi Arabia, Iran, USA)	Nominal
	Parent responsi-ble for student	Student's parent such as (father or mum)	Nominal
Academic Background Features	Stage ID	Stage Student belongs to such as (Low level, Middle level, High level)	Nominal
	Grade ID	Grade students belongs such as (G-01, G-02, G-03, G-04, G-05, G-06, G-07, G-08, G-09, G-10, G-11, G-12)	Nominal
	Section ID	Classroom student belongs to such as (A,B,C)	Nominal
	Semester	School year semester such as (First or second)	Nominal
	Topic	Course topic such as (Math, Eng-lish, IT, Arabic, Science, Quran)	Nominal
	Student Absence Days	Student absence days (Above-7, Below-7)	Nominal
Parents Partici-pation on learning pro-Cass	Parent Answering Survey	Parent is answering the surveys that provided from school or not	Nominal
	Parent School Satisfaction	This attribute obtains the degree of parent satisfaction from school as follow (Good, Bad)	Nominal
Behavioral Features	Discussion groups	Student behavior during interac-tion with Kalboard 360 e-learning System.	Numeric
	Visited resources		Numeric
	Raised hand on class		Numeric
	Viewing an-nouncements		Numeric

5 CONCLUSION

This research work has studied the challenges associated with predicting the academic performance of students using a Kalboard 360 educational dataset. The accurate prediction of student performance can help tutors and instructors to identify and concentrate more on students that are failing or likely to fail, therefore providing assistance for them. Classification techniques have been utilized in order to enhance the outcome of student performance. Furthermore, a new boosting algorithm with higher accuracy with so much efficiency has been introduced which is known as ADDE. The AdaBoostM2 ensemble technique was used with DE and the outcome illustrates that the use of DE with boosting technique produces better and desired optimized solution by reducing the weak learners and increasing the prediction accuracy. We made comparisons of our proposed algorithm with some previous works of classification algorithms that made use of same datasets as ours, this is shown in Table 6. The outcome showed that our proposed algorithm outshines other algorithms.

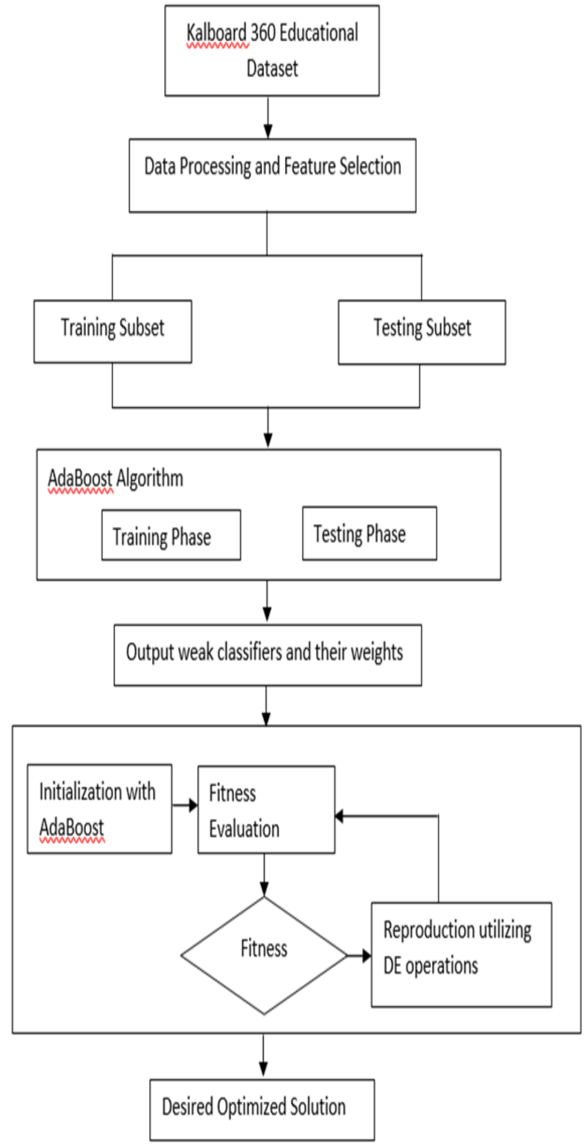

FIG. 3. *THE STRUCTURE OF THE MODEL ADDE*

REFERENCES

[1] D. M. D. Angeline, P. Ramasubramanian and I.S. Peter, Increased Success Outcome Of A Learner With Ensemble Teaching And Analysis With Naive Bayes Algorithm (2017).
[2] H. M. Harb and M. A. Moustafa, Selecting Optimal Subset of Features for Student Performance Model (2018).
[3] S. Fong, R.P. Biuk-Aghai and R.C, Swarm Search Methods in Weka for Data Mining. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing (pp. 122-127) [2012] ACM.
[4] A. Daud, N.R. Aljohani, R. A. Abbasi, M. D. Lytras, F. Abbas and J.S. Alowibdi, Predicting student performance using advanced learning analytics. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 415-421), (2017). International World

Wide Web Conferences Steering Committee.
[5] M. Ramaswami and R. Bhaskaran, A study on feature selection techniques in educational data mining. arXiv preprint arXiv:0912.3924 (2009).
[6] E. A. Amrieh, T. Hamtini and I. Aljarah, Preprocessing and analyzing educational data set using X-API for improving student's performance. In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT) (pp. 1-5), (2015), IEEE.
[7] R. Asif, A. Merceron, S. A. Ali and N. G. Haider, Analyzing undergraduate students' performance using educational data mining. Computers & Education, 113, 177-194 (2015).
[8] Y. Freund and R. Schapire, Experiments with a New BoostingAlgorithm, Proceedings: ICML'96, 148–156 (1996).
[9] S. S. Ajibade and A. Adediran, An overview of big data visualization techniques in data mining. International Journal of Computer Science and Information Technology Research, 4(3), 105-113 (2016).
[10] A. S. ElDen, M. A. Moustafa, H. M. Harb and A. H. Emara, AdaBoost ensemble with simple genetic algorithm for student prediction model. International Journal of Computer Science & Information Technology, 5(2), 73, (2013).
[11] Oh, Dong-Yop Oh, "GA-BOOST: A Genetic Algorithm for Robust Boosting",A PhD thesis presented to the University of Alabama,Tuscaloosa, Alabama (2012).
[12] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in the proceedings of the13th International Conference on Machine Learning,Italy, pages 148 -156 (1996).
[13] R. Storn & K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359, (1997).
[14] M. Pandey and S. Taruna, A comparative study of ensemble methods for students' performance modeling. International Journal of Computer Applications, 103(8), (2014).
[15] M. Stapel, Z. Zheng and N. Pinkwart, An Ensemble Method to Predict Student Performance in an Online Math Learning Environment. International Educational Data Mining Society (2016).
[16] M. H. Rahman and M. R. Islam, Predict Student's Academic Performance and Evaluate the Impact of Different Attributes on the Performance Using Data Mining Techniques. In 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE) (pp. 1-4), 2017. IEEE.
[17] L. Rahman, N. A. Setiawan and A. E. Permanasari, Feature selection methods in improving accuracy of classifying students' academic performance. In 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)(pp. 267-271). 2017, IEEE.
[18] O. W. Adejo and T. Connolly, Predicting student academic performance using multi-model heterogeneous ensemble approach. Journal of Applied Research in Higher Education, 10(1), 61-75, 2018.
[19] M. M. Alam, K. Mohiuddin, A. K. Das, M. K. Islam, M. S. Kaonain and M. H. Ali, A Reduced feature based neural network approach to classify the category of students. In Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence (pp. 28-32). 2018, ACM.
[20] Bendangnuksung and P. Prabu, Students' Performance Prediction Using Deep Neural Network. International Journal of Applied Engineering Research Volume 13, Number 2. (pp. 1171-1176), 2018.