Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **3**, September **2023**

An Educational Data Mining System for Predicting Students' Programming Performance

Md. Rashedul Islam*,[a], Adiba Mahjabin Nitu[b], Dr. Md. Abdulla Al Mamun[c]
Md. Abu Marjan[d], Md. Palash Uddin[e], Masud Ibn Afjal[f]

Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University
Dinajpur-5200, Bangladesh

* Corresponding Author: rashed.itcell@hstu.ac.bd
[b]nitu.hstu@gmail.com, [c]mamun@hstu.ac.bd, [d]marjan@hstu.ac.bd
[e]palash_cse@hstu.ac.bd, [f]masud@hstu.ac.bd

Received April 2023; revised July 2023

Abstract. *Educational Data Mining (EDM) plays a vital role in discovering meaningful relationships from educational data. In recent years, the application of EDM has become an active research field for predicting the performance of students. In this paper, we propose an EDM paradigm for classifying the students' performance in programming more precisely over real collected data. In the proposed EDM system, we investigate an effective feature engineering process and an ensemble machine learning classifier to boost the prediction performance. The classification process identifies the current programming status of students into four groups: excellent, good, average, and weak. In particular, we analyze Random Forest (RF) classifier and measure the performance using accuracy, precision, recall, f1-score, RMSE, kappa-score, and ROC. We train the RF model over 1700 data samples and the experimental results show that the predicting accuracy is 94%.*
Keywords: Educational Data Mining, Machine Learning, Feature Engineering, Programming Skill, Data Mining

1. **Introduction.** Data mining is the procedure of identifying data patterns and classifying them into different classes in order to extract interpretable and useful information from a large dataset [1]. Applying Machine Learning (ML) and data mining methods in education is an emerging research field, referred to as Educational Data Mining (EDM) [2]. In particular, the EDM process converts the educational field's raw data into knowledge that has a significant effect on both educational research and practice [3]. As ML models' performance largely relies on data representation [4], predicting the students' performance in education management through EDM is a great concern. For instance, it could recommend a proper suggestion to the lower-class students predicting their grades, and help them overcome and solve all complications in their study. Nowadays, Computer Science (CS) related courses are quite popular among students. As such, competition is increasing rapidly in the job field of CS-related domains. It is considered that the performance of programming skills is the most important factor to success in this domain. The researchers in [5] predict that programming skills are the most efficient indicator of success in CS. Students can prepare themselves before entering professional life by improving their programming skills, which refer to a variety of skill-set, such as knowledge of different computer programming languages, Knowledge of algorithms, problem understanding

and solving capability, mathematical skills, technical skills, writing skills, creativity level, passion for learning code, etc. Several data mining research works in educational data processing have been conducted over the last decade, such as student behavior analysis, academic data analysis, dropout prediction, etc. [6] [7].

In recent years, EDM has become one of the active research fields and we discuss here the most related EDM research works. The authors in [8] predict the final result of the engineering students' by analyzing the first three-year results using regression algorithms. They achieve 89.15% accuracy through Linear Regression (LR). In [9], the researchers use DM techniques and video learning analytics to predict students' final performance and their experiment shows that Random Forest (RF) predicts accurately with an 88.3% accuracy level. The study in [10] present a ML model that helps stakeholders and students to choose a proper steam (science, humanities and commerce) for their higher studies. Authors find that, extreme gradient boosting and Support Vector Machine (SVM) algorithm's performance were superior among the regressors algorithms. The researchers in [11] present that the AdaBoost algorithm obtains a significant result to discover students' problems that they would suffer in their courses such as system analysis and design and mathematics. The authors use grades for those subjects as trainable attributes of the ML models. The SVM and Artificial Neural Network (ANN) are effective at assessing the performance in various engineering courses for individual students. However, the multi-linear regression model produces an acceptable output to forecast all students' performance [12]. ANN, Naive Bayes Classifier (NBC), Decision Tree (DT) (C4.5), SVM, and k-Nearest Neighbor (k-NN) classifiers were used in [13] to predict and improve the assessment procedure of students' performance assessment at the final examination. DT (C4.5) algorithms can predict more accurately than other models with a 90% accuracy level. A few research works have been presented to assess programming skills using the EDM strategy. For instance, [14] presents to predict the performance of undergrad students in the programming language *C*. They considered only 70 data samples and a few features for this research. In that work, students are divided into three categories poor, average, and good. Authors use the DT algorithm to classify them and DT achieves 87% accuracy. If they would take more features and samples in the training dataset, the prediction result might be better. Another researcher in [15] predicts the student's programming skills for the tertiary level, providing an enhancing mechanism to develop programming skills. Authors assess some key ML classifiers to predict the performance in programming and RF shows the best result.

However, the current EDM approach in this domain has some drawbacks, such as researchers do not use any effective feature engineering process. They just follow a simple ML strategy directly. As such, in our proposed EDM system, we investigate an effective feature engineering process to improve prediction performance in programming skills classification. The real collected dataset used in this study is obtained from [15]. This dataset contains more than 1700 data samples with 36 features. We use the RF ensemble classifier for training, and testing the data samples. RF predicts the performance of the students in programming into four categories: excellent, good, average, and weak. To what follows, the two key contributions of our work are as follows:

- Investigation of an effective feature engineering process to enhance the prediction accuracy, and
- Analysis and investigation of the RF classifier for predicting the students' programming performance.

The rest of this article is organized as follows. Section 2 represents the overall approach of the proposed EDM system and the methods and materials including the prepossessing

of the dataset and ML classifier. The experiments and classification results are provided in Section 3. To the end, Section 4 summarizes and concludes the findings and observations.

2. Proposed EDM Approach.

2.1. **Overview of Proposed Methodology.** In today's world, technology is growing faster and CS is one of the most demanded subjects for students. In the field of CS, programming skills are a major concern. As such, students have to improve their programming skills to present themselves as an expert professional. From this motivation, we present our research to classify the students' performance in programming more precisely. We collect the dataset from existing research. The performance of ML models is largely reliant upon the data representation. Therefore, we investigate an effective feature engineering scheme to represent the dataset more interpretable to the ML model. Then, data imbalance is a common problem in ML. We use the SMOTE data balancing technique to balance the dataset. After preprocessing the dataset, we get an ML-trainable dataset, which is then split into training and testing sets. Then, the trained dataset is trained using the RF ensemble ML classifier. Finally, we evaluate the performance of the testing dataset using the classical performance measurement metrics (accuracy, precision, recall, f1-score, RMSE, kappa-score, and ROC). All experiments of this work are carried out using the Python programming language. To this end, Figure 1 presents the overview of the proposed EDM system.

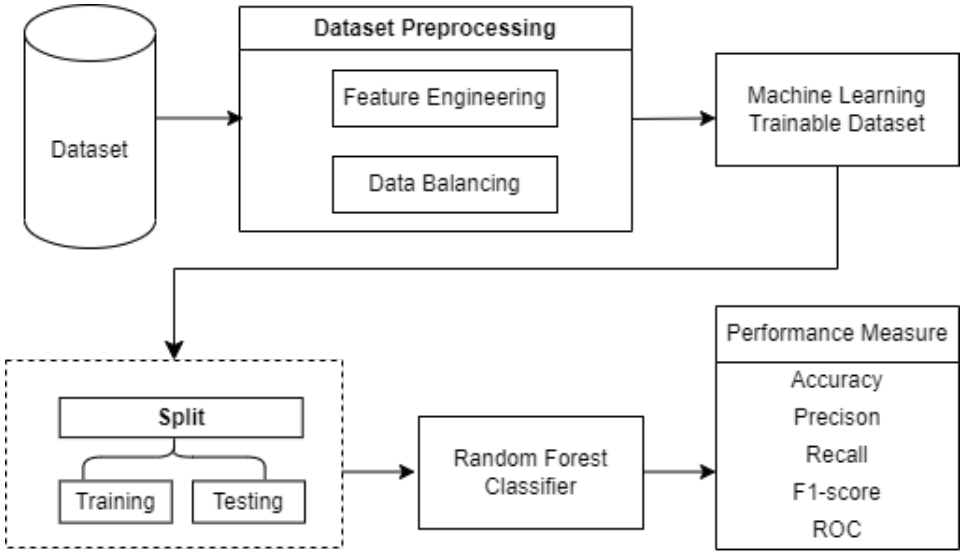

FIGURE 1. Overview of the proposed EDM system.

2.2. **Dataset Preprocessing.** The dataset of this work contains over 1700 data samples and 36 features for which we design an effective feature engineering process. To begin with, we balance the imbalanced data using SMOTE. Table 1 represents the result of applying SMOTE to the dataset. Then, we investigate an effective data representation technique. We apply both One Hot Encoding and Label Encoding techniques to transform the categorical values into numerical values for enhancing the data understanding of the model. To do this, at first we calculate the number of unique values for each feature. After a few trials, we apply One Hot Encoding on those features containing 3-5 unique

values and apply Label Encoding to other features. This leads to our final ML-trainable dataset.

SMOTE: SMOTE is an oversampling method widely used in ML for imbalance data handling with high-dimensional data [16]. In order to increase the features or instances number, the SMOTE technique creates randomly new instances or samples of the minority class instances from the line connecting the minority class of instances nearest neighbors.

One Hot Encoding: One Hot Encoding makes the data more understandable to the model [17]. For each distinct level of a categorical attribute or feature, a new variable is created in this encoding technique and each class or category is mapped to a binary variable that can either be 0 or 1. Thus, 0 is referred to as the absence of that class or category and 1 is referred to as its presence [18].

Label Encoding: In Label encoding, retaining the order is crucial. As a result, the encoding ought to reflect the sequence. Each label is mapped into an integer value in label encoding technique [18].

Table 1. Applying SMOTE to the dataset.

	Excellent	Good	Average	Weak
Before applying SMOTE	564	307	346	503
After applying SMOTE	564	564	564	564

2.3. **ML Model.** After preparing the ML trainable dataset, we split this dataset for training and testing. Then, we train this dataset with the RF ensemble classifier to predict students' programming performance. We try different hyperparameter values for the RF classifier and we use optimal hyperparameter values n_estimators = 100, random_state=42.

Random Forest: RF is a decision trees-based ensemble ML learning algorithm [19] that takes a number of decision trees on various subsets from the main dataset and takes the average result to increase the predictive accuracy of the given dataset. This algorithm can predict categorical as well as continuous data using classification and regression methods. Each decision tree uses a simple deterministic probability to select randomly the significant relevant feature of data samples and takes randomly the subset of the given dataset as ML trainable data [20]. Consecutively, the data is split into the relatively same sets for each tree which are simply denoted as nodes, to enhance the prediction accuracy of testing data. These dividing nodes are identified using the predictor vector value. Vectors that split the datasets are sometimes named critical vectors. The RF classifier fits a variety of predefined bootstrapped datasets on various decision trees. The mode of the classes from all decision trees is a categorical response of the predicted value. The predicted result is the average value of the fitted response of an uninterrupted response from all the independent trees of each bootstrapped sample. Simply, instead of depending on one decision tree, it takes the prediction from the individual tree, and on the basis of averaging or majority votes of predictions, RF predicts the final output [21].

2.4. **Model Evaluation.** To evaluate the performance of the RF classifier for predicting students' programming performance, we employ the key widely used performance measurement metrics, which are accuracy, precision, F1-score, and recall [22]. In an ML model accuracy is referred to as the testing accuracy [23], which states the percentage of the dataset's actual value that approves with the predicted value, and it helps to classify the students. Precision calculates the positive predictive value or probability of a positive test result [24]. As we categorize the students' according to their performance in programming, precision specifies that the percentage among them those are exactly

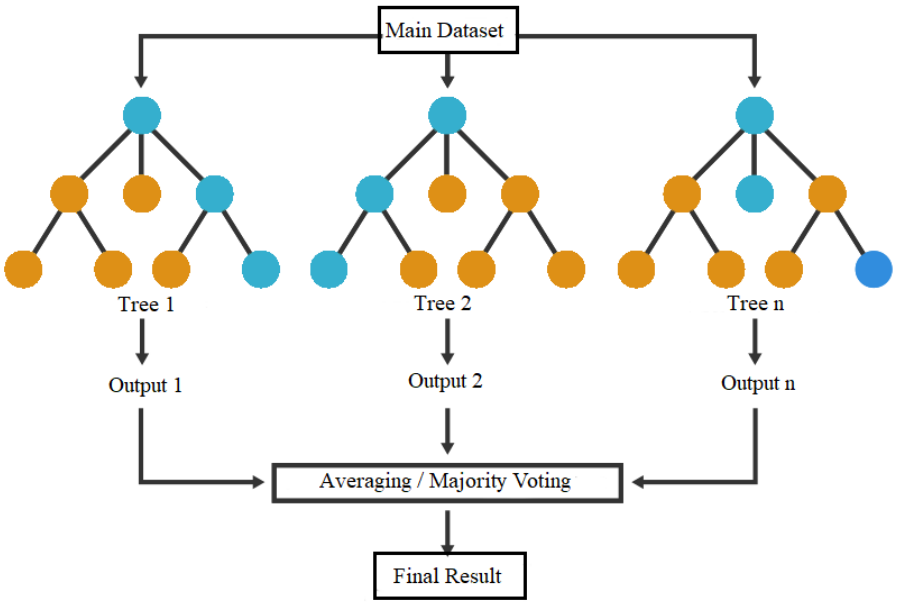

FIGURE 2. Working process of RF classifier in the top-down approach.

classified into the specified categories of students (excellent, good, average, and weak). Recall specifies the probability value of true positives (TP) from total predicted positive values by the ML model [25]. F1-score practices to make fast actions concerning related methods. The value of the F1-score is gained from recall and precision. In the imbalanced dataset, sometimes precision, F1-score, and recall are assumed as more effective performance measurement metrics than accuracy [26]. ML model produces True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values while training. Performance measurement tools of ML are constructed using TP, TN, FP, FN values as follows [27]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\_Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

3. **Result and Discussion.** We perform experiments in two steps. At first, we split the ML trainable dataset into a 70:30 ratio for training and testing. In the next step, we divide the ML trainable dataset into a 60:40 ratio for training and testing. In the first experiment, RF achieves 94% accuracy, and in the next experiment with a 60:40 ratio RF achieves 93% accuracy. Table 2 and Table 3 represent average accuracy for all classes and precision, recall, and f1-score for individual classes of RF classifier for 70:30 and 60:40 dataset split, respectively. In the first experiment, the average result of precision, recall, and f1-score is 94%, and in the next experiment with a 60:40 ratio average value of precision, recall, and f1-score is 93%.

In addition, we use RMSE, Cohen's kappa coefficient and the ROC curve to present how the model is fit between the model and the dataset. The value of RMSE shows how

TABLE 2. Classification results, rmse score, and kappa score of RF classifier for 70:30 dataset split.

Class	Accuracy	Precision	Recall	F1-score	RMSE	Kappa-score
Excellent	0.94	0.93	0.94	0.93	0.26	0.93
Good		0.93	0.96	0.97		
Average		0.97	0.94	0.93		
Weak		0.93	0.92	0.92		

TABLE 3. Classification results, rmse score, and kappa score of RF classifier for 60:40 dataset split.

Class	Accuracy	Precision	Recall	F1-score	RMSE	Kappa-score
Excellent	0.93	0.92	0.91	0.90	0.30	0.92
Good		0.93	0.96	0.97		
Average		0.95	0.92	0.93		
Weak		0.93	0.92	0.92		

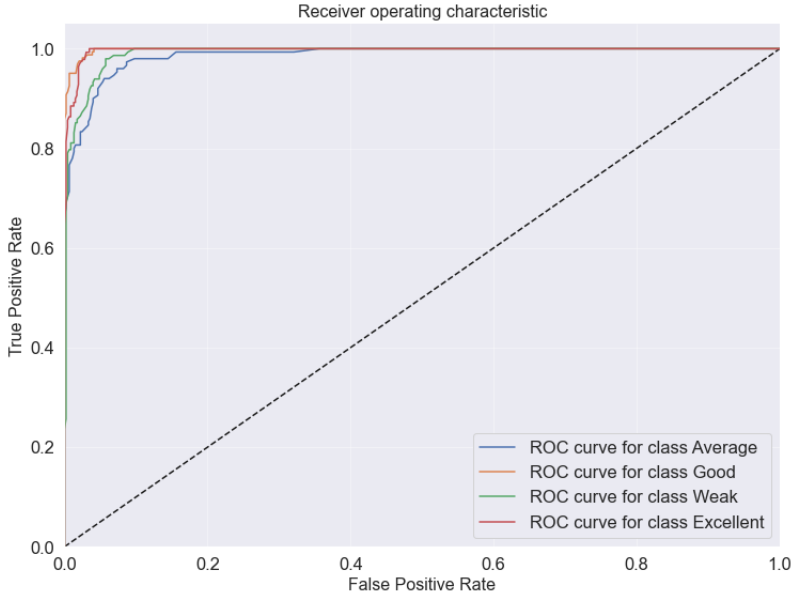

FIGURE 3. ROC curve for 70:30 dataset split.

significantly a trained model fits the testing dataset. Cohen's kappa coefficient represents the relationship between investigational accuracy and expected accuracy. From Table 2 and Table 3, we can see that RMSE and Kappa-score for all classes best fit for 70:30 dataset split and the value of RMSE and Kappa-score are 26% and 93%. ROC curve represents the relationship of the TP rate vs the FP rate [15]. Fig. 3 and Fig. 4 shows the ROC curve for the 70:30 and 60:40 dataset split, respectively. Finally, Fig. 3 shows that the ROC curve is best fitted for all classes.

Comparison with the Existing Works: To this end, from Table 4, we can see that the EDM studies in this domain obtain 80-91% accuracy while in our proposed EDM system RF obtain 94% accuracy which is better than previous research works in this domain. We only consider the best classifier in each case and finally compare it with our proposed method. The proposed EDM outperforms all the models in terms of accuracy.

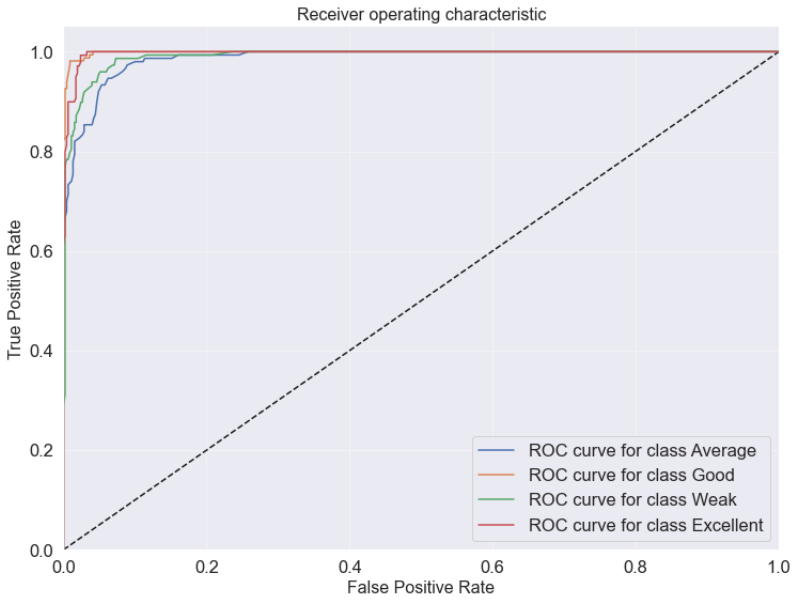

FIGURE 4. ROC curve for 60:40 dataset split.

TABLE 4. Works on existing literature and comparison with the proposed EDM system.

Method	Objective	Classification Techniques	Best Classifier (Accuracy)
Adekitan et al. [8]	Analysis of the performance of engineering students	Regression models	LR (89%)
Hasan et al. [9]	Predicting Higher Educational Institutions students' performance	RF, LR, NBC, SVM	RF (88%)
Huang et al. [12]	Evaluating Performance in an Engineering Course	ANN, MLR and SVM	ANN (89%)
Livieris et al. [13]	Improving the assessment procedure of students' performance	ANN, NBC, SVM, DT and k-NN	DT (90%)
Pathan et al. [14]	Predicting undergrad student's performance in programming C	DT (ID3)	DT (87%)
Marjan et al. [15]	Classification of tertiary students' programming skill	DT, SVM, ANN, RF, NBC, k-NN	RF (91%)
Proposed EDM	Predict students' performance in programming	RF	RF (94%)

4. **Conclusion and Future Work.** In this paper, we have proposed an EDM system that predicts students' performance in programming more accurately than previous models. To do this, we investigate an effective feature engineering process. To classify students' performance, we train the dataset with an RF classifier and to evaluate the performance, we perform our experiments with two training and testing ratios. All experimental results show that the RF classifier obtains a 94% accuracy level to predict the student's performance in computer programming. In this research, we do not present how the model works for individual classification of the dataset. In the future, we could add explainability to explore important features responsible for the classification process, which

would be helpful for recommendations for the students to improve their performance in programming.

REFERENCES

[1] H. A. Madni, Z. Anwar, and M. A. Shah, "Data mining techniques and applications—a decade review," in *2017 23rd international conference on automation and computing (ICAC)*. IEEE, 2017, pp. 1–7.

[2] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1355, 2020.

[3] W. Xiao, P. Ji, and J. Hu, "A survey on educational data mining methods used for predicting students' performance," *Engineering Reports*, vol. 4, no. 5, p. e12482, 2022.

[4] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021.

[5] S. A. Ahmed, M. A. Billah, and S. I. Khan, "A machine learning approach to performance and dropout prediction in computer science: Bangladesh perspective," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2020, pp. 1–6.

[6] D. Buenaño-Fernandez, W. Villegas-CH, and S. Luján-Mora, "The use of tools of data mining to decision making in engineering education—a systematic mapping study," *Computer applications in engineering education*, vol. 27, no. 3, pp. 744–758, 2019.

[7] I. Menchaca, M. Guenaga, and J. Solabarrieta, "Using learning analytics to assess project management skills on engineering degree courses," in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2016, pp. 369–376.

[8] A. I. Adekitan and O. Salau, "The impact of engineering students' performance in the first three years on their graduation result using educational data mining," *Heliyon*, vol. 5, no. 2, p. e01250, 2019.

[9] R. Hasan, S. Palaniappan, S. Mahmood, A. Abbas, K. U. Sarker, and M. U. Sattar, "Predicting student performance in higher educational institutions using video learning analytics and data mining techniques," *Applied Sciences*, vol. 10, no. 11, p. 3894, 2020.

[10] S. Ahmad, M. G. R. Alam, J. Uddin, M. R. Bhuiyan, and T. S. Apon, "Machine learning based stream selection of secondary school students in bangladesh," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 11, no. 1, pp. 105–118, 2023.

[11] K. Kaur and K. Kaur, "Analyzing the effect of difficulty level of a course on students performance prediction using data mining," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2015, pp. 756–761.

[12] S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," *Computers & Education*, vol. 61, pp. 133–145, 2013.

[13] I. E. Livieris, T. Kotsilieris, V. Tampakas, and P. Pintelas, "Improving the evaluation process of students' performance utilizing a decision support software," *Neural Computing and Applications*, vol. 31, pp. 1683–1694, 2019.

[14] A. A. Pathan, M. Hasan, M. F. Ahmed, and D. M. Farid, "Educational data mining: A mining model for developing students' programming skills," in *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*. IEEE, 2014, pp. 1–5.

[15] M. A. Marjan, M. P. Uddin, and M. Ibn Afjal, "An educational data mining system for predicting and enhancing tertiary students' programming skill," *The Computer Journal*, 2022.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," 2018.

[18] S. Saxena, "Here's all you need to know about encoding categorical data (with python code)," Jun 2022.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[20] M. M. I. Raju, S. Sarker, and M. M. Islam, "Chronic kidney disease prediction using ensemble machine learning," 2023.

[21] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," *Ensemble machine learning: Methods and applications*, pp. 157–175, 2012.

[22] T. Devasia, T. Vinushree, and V. Hegde, "Prediction of students performance using educational data mining," in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, 2016, pp. 91–95.

[23] R. M. De Albuquerque, A. A. Bezerra, D. A. de Souza, L. B. P. do Nascimento, J. J. de Mesquita Sá, and J. C. do Nascimento, "Using neural networks to predict the future performance of students," in *2015 International Symposium on Computers in Education (SIIE)*. IEEE, 2015, pp. 109–113.

[24] K. Kaur and K. Kaur, "Analyzing the effect of difficulty level of a course on students performance prediction using data mining," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2015, pp. 756–761.

[25] S. Annamalai, R. Udendhran, and S. Vimal, "An intelligent grid network based on cloud computing infrastructures," in *Novel practices and trends in grid and cloud computing*. IGI Global, 2019, pp. 59–73.

[26] H. Wu, S. Yang, Z. Huang, J. He, and X. Wang, "Type 2 diabetes mellitus prediction model based on data mining," *Informatics in Medicine Unlocked*, vol. 10, pp. 100–107, 2018.

[27] I. M. K. Karo, M. F. M. Fudzee, S. Kasim, and A. A. Ramli, "Sentiment analysis in karonese tweet using machine learning," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 1, pp. 219–231, 2022.