IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 14, 2021, accepted October 6, 2021, date of publication October 13, 2021, date of current version October 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119596

Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks

AYA NABIL, MOHAMMED SEYAM, AND AHMED ABOU-ELFETOUH
Department of Information Systems, Faculty of Computers and Information Science, Mansoura University, Mansoura 35516, Egypt
Corresponding author: Aya Nabil (ayanabil@mans.edu.eg)

ABSTRACT Predicting students' academic performance at an early stage of a semester is one of the most crucial research topics in the field of Educational Data Mining (EDM). Students are facing various difficulties in courses like "Programming" and "Data Structures" through undergraduate programs, which is why failure and dropout rates in these courses are high. Therefore, EDM is used to analyze students' data gathered from various educational settings to predict students' academic performance, which would help them to achieve better results in their future courses. The main goal of this paper is to explore the efficiency of deep learning in the field of EDM, especially in predicting students' academic performance, to identify students at risk of failure. A dataset collected from a public 4-year university was used in this study to develop predictive models to predict students' academic performance of upcoming courses given their grades in the previous courses of the first academic year using a deep neural network (DNN), decision tree, random forest, gradient boosting, logistic regression, support vector classifier, and K-nearest neighbor. In addition, we made a comparison between various resampling methods to solve the imbalanced dataset problem, such as SMOTE, ADASYN, ROS, and SMOTE-ENN. From the experimental results, it is observed that the proposed DNN model can predict students' performance in a data structure course and can also identify students at risk of failure at an early stage of a semester with an accuracy of 89%, which is higher than models like decision tree, logistic regression, support vector classifier, and K-nearest neighbor.

INDEX TERMS Deep neural networks, educational data mining, imbalanced dataset problem, machine learning, predicting students' performance, resampling methods.

I. INTRODUCTION

Education plays an important role in the progress of a nation. It is also a crucial tool for success in life. Any educational institution tries to provide good education to its students to improve the learning process [1]. The academic performance of students is an essential factor that influences the accomplishment of any educational institution. During the learning process at different levels of education, the failure rates and dropouts of computer programming courses are two essential problems faced by students [2], [3].

Artificial intelligence (AI) and machine learning (ML) have been applied in various fields such as image classification, natural language processing, speech recognition, text translation, and the field of educational data mining (EDM). EDM is concerned with applying various data mining techniques such as classification, regression, time series analysis, and association rule mining in the education field to analyze and evaluate various aspects of educational datasets collected from different e-learning environments or higher educational institutions. EDM is one of the most common techniques used to develop predictive models to extract hidden patterns and useful information, which can help in education and learning [4].

Educational institutions have started to apply AI technology to enhance the learning process of students [5]. Today, educational institutions have an important challenge in providing high-quality education to their students and enhancing their success rate [6]. ML plays an important role in the education field for predicting students' academic performance in the future and helps students to achieve higher grades [7]. It is essential to predict the academic success of students because it is a crucial process to determine the students who have a risk of failure at an early stage of a

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Lai.

semester assessment. Therefore, these students will be given some remediation to increase their academic achievements before the final evaluation and to increase the success rate of the university [8].

Various data mining techniques have been used to predict different educational outcomes, such as performance, achievement, retention, dropout rate, and success [9]. Data mining techniques are extremely helpful in the education field, especially for analyzing and predicting students' academic performance.

Predicting students' academic performance at an early stage of a semester is a very useful tool for taking early actions to enhance their performance and also to reduce the failure rates of students at the end of a semester. However, predicting students' academic performance is a serious challenge because different features can affect the performance of students, such as academic background, which are the previous academic achievements, demographic features, economic background, behavioral features, and other factors. Hence, EDM is an important tool for solving this problem [10]. Using historical academic data of students to predict their future performance is considered one of the most common applications of EDM. It is an essential tool that can be used to enhance students' performance, reduce failure rates, and provide a complete picture of the learning process of students [11], [12].

Nowadays, educational institutions are generating massive amounts of educational data, and these data are used for data analytics for the decision-making process to enhance the performance of students. This may lead to an improvement in overall educational settings and a better understanding of the learning process [13], [14].

One of the most important factors affecting the performance of classifiers is the imbalanced dataset problem. It is a severe challenge that appears in the field of EDM and leads to misleading results and poor performance. Many resampling techniques have been developed to handle imbalanced classes. Hence, the method proposed in this paper aims to address the imbalanced class and how we can handle this problem using various resampling methods such as SMOTE, ROS, ADASYN, and SMOTE-ENN to improve the performance of the models and to achieve reliable results.

In this study, a dataset collected from a public 4-year university is used to develop our predictive models using five algorithms: deep artificial neural network (DNN), decision tree (DT), logistic regression (LR), support vector classifier (SVC), K-nearest neighbor (KNN) random forest (RF), and gradient boosting (GB) to analyze and evaluate the students' data to predict students' success in a Data Structure course and to identify at-risk students at an early stage of a semester based on their grades in the previous courses of the first academic year. We used the main steps of the knowledge discovery in databases (KDD) process in our studies, such as data collection, data pre-processing, data mining process, and performance evaluation.

From the experimental results, it is observed that the proposed DNN outperformed the others in terms of accuracy, recall, F1-score, and classification error metrics, while SVC outperformed the others in terms of precision. Moreover, the proposed model may be used as a tool for early prediction of students at risk of failure at an early stage of a semester; thus, this early-stage prediction helps to better advise students for failure prevention and improve the overall learning process.

The processes of our research study are as follows:

- Collecting the dataset and performing data pre-processing tasks to achieve better results.
- Various resampling methods such as SMOTE, random over-sampling, ADASYN, and SMOTE-ENN were applied to handle the imbalanced dataset problem.
- Various model validation methods, namely random hold-out, and stratified 5-fold cross-validation were applied.
- Some machine learning techniques, such as deep artificial neural network (DNN), decision tree (DT), logistic regression (LR), support vector classifier (SVC), K-nearest neighbor (KNN), random forest (RF), and gradient boosting (GB) were applied on various balanced datasets using the resampling methods defined above.
- Evaluating the performance of the models using various evaluation methods, such as accuracy, precision, recall, F1-Score, and classification error.
- Comparing the performance of the classifiers defined above and choosing the best one.
- Using a statistical hypothesis test for the comparison of the examined methods.

The remainder of this paper is organized as follows: Section 2 provides an overview of some related works, Section 3 describes our proposed methodology, Section 4 discusses the experimental results and discussion, and finally, Section 5 discusses the conclusion and future work.

II. RELATED WORKS

Students' academic performance is one of the most important factors in higher education. Several researchers have used EDM applications to predict and evaluate students' academic performance in the decision-making process and to understand the learning process [4]. In this section, we discuss previous studies based on online and real datasets for predicting students' academic performance using different ML techniques. Table 1 shows a comparison of previous studies in predicting students' academic performance.

We have performed a literature review on predicting students' academic performance using different machine learning techniques, and it was observed that the CGPA and GPA of students are the most common indicators used as the predicted values for evaluating and predicting students' academic achievement at the university level [15]–[19]. Some researchers have used other attributes such as quiz grades, midterm marks, assessments, attendance, and lab work in their works to predict students' academic performance [20], [21].

TABLE 1. Comparison of some previous works in predicting students' academic performance.

Paper	Dataset Size	ML Algorithms	Best Algorithm	Metrics	DM Task	Output Feature	Source Data	Year
[22]	592	Linear Regression, Random Forest, Bagging, Gaussian Processes, M5, and M5' rules	Random Forest	Mean Absolute Error (MAE)	Regression	-	Educational Institution (Real Data)	2018
[23]	550	Support Vector Classifier (SVC), and Naïve Bayes (NB).	SVC with an accuracy of 87%	Accuracy, Precision, and Recall	Classification	{Pass, Fail}	Nizwa University (Real Data)	2020
[24]	203	Classification Association Rule Mining (CARM)	CARM with an accuracy of 67.33%	Accuracy	Classification and Association Rule Mining	{Good, Bad}	Mathematics Department (Real Data)	2016
[25]	500	Convolutional Neural Network (CNN)	CNN	Accuracy, Precision, Recall, F1-Measure	Classification	{Low, Medium, High}	Online Data (Kaggle)	2020
[26]	500	ANN, Decision Tree, Random Forest, Bagging, Voting, and Boosting	81.18% F1-score of Random Forest with Genetic algorithm	Geometric-Mean, F1-score, Area Under Curve (AUC), Precision, True Positive Rate, F1-Measure, True Negative Rate	Classification	{Low, Medium, High	Online Data (Kaggle)	2020
[27]	500	MLP, Decision Tree, Random Forest, Naïve Bayes, IBK, Decision Table, REP Tree, and Random Tree	MLP with an accuracy of 78.33%	Accuracy, Sensitivity, F1-measure, Specificity, Kappa statistic, Time, Roc curve, and RMSE Error	Classification	{Low, Medium, High}	Online Data (Kaggle)	2019
[28]	500	Decision Tree and K-Nearest Neighbor	Decision Tree with an accuracy of 71.09%.	Accuracy, Precision, and Recall	Classification	{Low, Medium, High}	Online Data (Kaggle)	2020
[29]	500	Decision Tree, Logistic Regression, Naïve Bayes, and some ensemble algorithms like Bagging, Random Forest, Voting, and Boosting	Boosting with an accuracy of 75%	Accuracy	Classification	{Low, Medium, High}	Online Data (Kaggle)	2021
[30]	500	Naïve Bayes, Decision Table, MLP, and J48 Ensemble Methods like Bagging, RandomSubSpace, and AdaBoost	AdaBoost with MLP technique with an accuracy of 80.33%	Accuracy	Classification	{Low, Medium, High}	Online Data (Kaggle)	2021
[31]	649	ANN, Bagging and Boosting	Bagging with an accuracy of 88%	Accuracy, Precision, Recall, False Positive Rate, F1-Measure, True Positive Rate, and Confusion Matrix	Classification	N/A	Online (UCI Machine Learning Repository)	2021

Some researchers have used students' academic achievements in previous courses to predict their performance in upcoming courses. Some traditional ML techniques have been used to predict students' grades in upcoming courses and to identify at-risk students at an early stage of a semester [22]–[24]. In [22], the authors carried out some

TABLE 1. ***(Continued.)*** **Comparison of some previous works in predicting students' academic performance.**

[32]	1073	Deep Dense Neural Network (DDNN), Decision Tree, KNN, MLP, SGD, Random Forest, and NB	DDNN with an accuracy of 78-81%	F1-Measure and Area Under Curve (AUC)	Classification	{Pass, Fail}	Open University	2020
[33]	44	ANN and Naïve Bayes	ANN	F1-Measure, Precision, Recall, TP Rate, and FP Rate	Classification	{Graduated, Not-Graduated}	Educational Institution (Real Data)	2020
[34]	6807	Random Forest, Logistic Regression, Support Vector Classifier, Voting, Decision Tree, Bagging, MLP, and AdaBoost	93.8% F1-score of Random Forest	Precision, Recall, F1-Measure	Classification	{complete, withdrawn}	Technical institute (Real Data)	2021
[35]	388	Random Forest, Logistic Regression, and K-Nearest Neighbor	Random Forest with an accuracy of 93%	Accuracy	Classification	{submitted, not submitted}	Virtual Learning Environment	2021

experiments using various regression methods to predict the students' grades in the courses of the second semester given the grades of the courses in the first semester and demographic features, whereas the authors of [23] built a model to predict students' grades in a math course in the second semester based on their previous grades from school and their grades in the previous courses of the first semester using Support Vector Classifier and Naïve Bayes, whereas the paper of [24] Classification based on association rule mining has also been used to predict students' grades in a programming course given the students' grades in previous courses such as English and mathematics. It was observed that the dataset used in the previous works described above was small. In addition, predictive models are built using only traditional ML techniques.

Therefore, we will explore the efficiency of DNN and some traditional ML techniques to analyze and evaluate the students' data gathered from a public 4-year university to predict the success rate of students in a data structure course and to identify at-risk students at an early stage of a semester based on their grades in the previous courses of the first academic year.

In [25]–[30], an online dataset collected from a learning management system called Kalboard 360 was used to predict students' performance. This dataset contains 500 records and 16 features, The authors of [25] applied a Convolutional Neural Network algorithm (CNN) to explore the efficiency of Deep Learning in predicting students' performance, whereas the authors of [26] applied some ML techniques like ANN, Decision Tree, Random Forest, Bagging, Voting and Boosting using a Genetic Algorithm for the feature selection process to enhance the models' performance, whereas the authors of [27] applied MLP and some traditional ML techniques like Decision Tree, Random Forest, Naïve Bayes to evaluate their performance, whereas the authors of [28] applied Decision tree and K-Nearest Neighbor using the SMOTE as an oversampling method to handle the problem of the imbalanced dataset to achieve better and reliable results, whereas the authors of [29] plotted different graphs to determine the important features which affect the performance of students and applied various classification algorithms such as Decision Tree, Logistic Regression, Naïve Bayes, and some ensemble algorithms like Bagging, Random Forest, Voting, and Boosting, whereas the authors of [30] applied some ML algorithms such as Naïve Bayes, Decision Table, MLP, and J48 and also they tried to improve the performance of these classifiers by using the ensemble methods like Bagging, RandomSubSpace and AdaBoost. It was observed that CNN achieved the best result and outperformed the others in terms of accuracy rate without applying any methods for the feature selection process because CNN made it automatically without human intervention.

In [31], another online dataset gathered from the UCI machine learning repository was used to predict students' academic performance in high school using different features such as academic background, personal attributes, and economic background. They tried to analyze the features and show the importance of using the economic background in the dataset, which affects the performance of students. Different ML techniques have been used, such as bagging, ANNs, and boosting. It was observed that the economic background plays an important role in the performance of students.

In [32]–[35], real datasets were used to identify students at risk of failure at an early stage of a semester. In [32], the authors built a model using a deep dense neural network and some traditional ML techniques such as decision tree, K-nearest neighbor, random forest, and naïve Bayes, and it showed that AUC and F1-score metrics are better than accuracy in the case of the imbalanced dataset. In [33], the authors built a model for predicting the probability of students' graduation on time, and it showed that artificial neural networks achieved better results than naive Bayes. In [34], the authors used different parameters (academic and non-academic) to demonstrate the importance of using non-academic parameters to predict students' performance. Random forest, logistic regression, support vector classifier, decision tree, bagging, MLP, and AdaBoost were used. It was observed that using non-academic parameters had a significant impact on the performance of students. The results achieved using all parameters were better than those obtained using only academic parameters. In [35], the authors built a model to predict students' performance in the next assignment submission based on various features such as final result, number of previous attempts, student credit, total clicks, student ID, age, and gender. Random forest, logistic regression, and K-nearest neighbor were applied in this study. In addition, they tried to find the most relevant features that affect the submission process of student assignments.

III. MATERIALS AND METHODS

The main goal of this paper is to develop predictive models using deep neural networks and some traditional machine learning techniques to predict students' academic performance in upcoming courses at an early stage of a semester based on their previous academic achievements. These models will be helpful because students will be informed of their probable results at an early stage of a semester. Therefore, they can increase their academic achievement at the end of the semester.

We used one of the most common online educational datasets that have been applied in most of the previous works, which is the students' academic performance dataset (xAPI-Edu-Data) collected from a learning management system called Kalboard 360. This dataset contains 500 records and 16 features. These features are categorized as follows demographic features, academic features, and behavioral features. At the beginning of our work in the field of EDM, especially in predicting students' academic performance, we used the educational dataset defined above to learn how to build a predictive model and how to use this model to predict students' performance using various features. We applied various ML algorithms such as decision tree, K-nearest neighbor, support vector machine, random forest, logistic regression, and multilayer perceptron. Then, some of the constructed models were applied to a real dataset to evaluate their performance.

The steps of the applied methodology are as follow:

A. DATA COLLECTION

In this experiment, we collected a real dataset of undergraduate students from a public 4-year university for a time of fourteen years (2006–2020). This dataset contains 4266 records of anonymized students with 12 features regarding their previous academic achievements during the first two academic years. These features are considered only academic features and are related to students' grades in the courses of the first academic year. The university gathered only academic features, and we tried to exploit these features to build a predictive model to be used by the university to reduce the number of failure rates. Most universities in various countries gathered only academic features, so our work is considered very important for them to use these previous academic grades. There are two semesters: one includes five courses, and the other includes six courses. The final score of each course was 100, and the completion of a particular course required a score of at least 50 in the final examination of that course. Each record in the dataset was described by the values of the 12 academic features. Table 2 lists the features of the dataset. Si, i =1, 2…11 correspond to the first two-semester course grades of the first academic year, and carryover of the first academic year (as CO_first_year), which is the number of courses that failed from previous semesters. Finally, the target feature, which is also a numeric variable, refers to the student's grade in the data structure course.

TABLE 2. Dataset description.

Attribute	Type	Description/ values
S1	Numeric	Physics
S2	Numeric	English
S3	Numeric	Mathematics
S4	Numeric	Calculus
S5	Numeric	Basics of Programming
S6	Numeric	Computer Science
S7	Numeric	Logic
S8	Numeric	Information System
S9	Numeric	Information Technology
S10	Numeric	Probability
S11	Numeric	OOP (Programming)
CO_first_year	Numeric	0,1,2
Target	Numeric	Data Structure (DS)

In this paper, we aim to predict the students' grades in a third semester's course in the second academic year, which is the data structure course as [pass, fail] given the students' grades in the previous courses of the first academic year.

We present a flow diagram of the proposed approach in Figure 1.

B. DATA PRE-PROCESSING

It is one of the most important steps in machine learning as it converts the raw data into a suitable format to solve the errors in the dataset collected from the real world and to achieve better results [19], [36]. We used the following steps: the data pre-processing step, which includes data cleaning, data discretization, feature encoding, handling imbalanced datasets, and data scaling. These steps are briefly described below.

1) DATA CLEANING

There are some missing values in our real data, such as the absence of a student in any exam. Therefore, we used this step to remove the missing values.

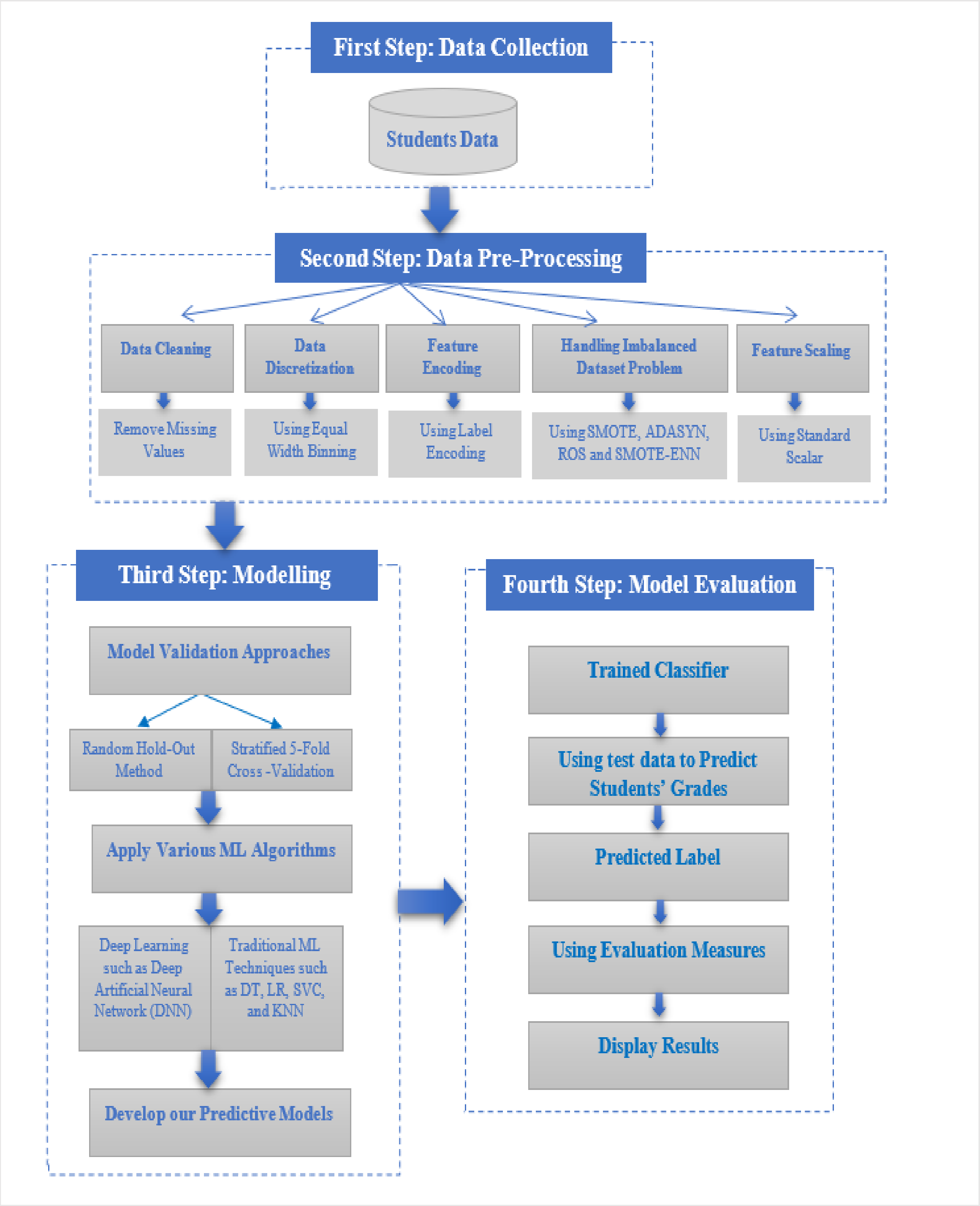
First Step: Data Collection
Students Data
Second Step: Data Pre-Processing
Data Cleaning
Data Discretization
Feature Encoding
Handling Imbalanced Dataset Problem
Feature Scaling
Remove Missing Values
Using Equal Width Binning
Using Label Encoding
Using SMOTE, ADASYN, ROS and SMOTE-ENN
Using Standard Scalar
Third Step: Modelling
Model Validation Approaches
Random Hold-Out Method
Stratified 5-Fold Cross -Validation
Apply Various ML Algorithms
Deep Learning such as Deep Artificial Neural Network (DNN)
Traditional ML Techniques such as DT, LR, SVC, and KNN
Develop our Predictive Models
Fourth Step: Model Evaluation
Trained Classifier
Using test data to Predict Students' Grades
Predicted Label
Using Evaluation Measures
Display Results

FIGURE 1. Flow diagram of the proposed approach.

2) DATA DISCRETIZATION

We used the discretization mechanism in this study to convert the numerical values of students' grades into nominal values to represent our dataset as a classification problem. We used the equal-width binning method to apply this step according to the standard grading of the university, as shown in Table 3. We divided the input features into five nominal intervals, namely, Excellent, Very Good, Good, Poor, and Fail based on the students' marks to make the algorithm learn easily to achieve better results. In addition, we divided the class label into two nominal intervals, Pass and Fail, based on students' grades. Table 3 presents the results for the discretization step.

We used the chi-square test as a statistical measure to identify the correlation between categorical variables using

TABLE 3. The result of the discretization step.

Input features	Student Marks	Class Label (Target)	Student Marks
Excellent	85-100	Pass	>=50
Very Good	75-84		
Good	65-74	Fail	<50
Poor	50-64		
Fail	< 50		

their frequent distribution. The chi-square test hypothesizes that there is no correlation between categorical variables. This test was based on a significance level of 0.05. If the hypothesis was rejected, then there was a statistical correlation between categorical variables. The chi-square equation (1) was used to identify the correlation between categorical variables, as described below.

$$X^2 = \sum \frac{(o-e)^2}{e} \tag{1}$$

where o is observed frequency and e is expected frequency.

After applying the chi-square test, we found that there was a significant correlation between the students' grades in each course of the first academic year and their academic performance in the upcoming course of the second year, which is the data structure course.

3) FEATURE ENCODING

At this stage, our features are categorical data, and machine learning algorithms cannot work with categorical data. Therefore, we need to convert these categorical data into a numerical form before feeding the input features into the model. In this study, we used an encoding technique called label encoding to convert our categorical data into a numerical form. This technique assigns an integer number to each distinct nominal variable. Our input features are converted into integer numbers. Excellent is 0, Very Good is 1, Good is 2, Poor is 3, and Fail is 4. In addition, the output feature was converted into integer numbers. Pass is 0, and Fail is 1.

4) HANDLING IMBALANCED DATASET

After applying the discretization step to our dataset, we observed a highly imbalanced dataset, and the distribution of the class label of students based on their grades was not equal. Our class label includes more samples for class "Pass," but the other class "fail" has fewer samples. The problem with a highly imbalanced dataset is shown in Figure 2. The distributions of the class label (pass, fail) were 91.16 % and 8.84 %, respectively. This is a severe challenge that appears in classification problems and leads to poor performance. Solving the problem of an imbalanced dataset is one of the most important factors for improving the performance of the models. This problem leads to the domination of the majority class over the minority class. Hence, the classifiers tend to be in the majority class and their performance is not reliable [19], [37]. Therefore, we need to solve this problem because this may lead to misleading results.

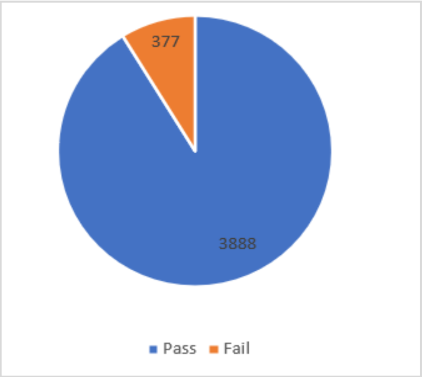

FIGURE 2. The distribution of the class label.

There are three categories of resampling methods for handling the problem of an imbalanced dataset: oversampling, undersampling, and hybrid sampling.

- Oversampling Method: This method generates new instances to increase the number of minority classes in the dataset to overcome the problem of an imbalanced dataset [38]. This is suitable for small data sizes. Some examples of oversampling techniques used in our study are the synthetic minority oversampling technique (SMOTE), adaptive synthetic sampling (ADASYN), and random oversampling (ROS) [39].
- Undersampling Method: It decreases the number of majority classes in the dataset to overcome the problem of an imbalanced dataset. This is suitable for a large amount of data. Edited nearest neighbor (ENN) is one of the most common undersampling methods.
- Hybrid Method: This technique is a combination of oversampling and undersampling techniques. SMOTE-ENN is an example of this technique that

In our study, we used different resampling methods to solve the problem of imbalanced datasets such as SMOTE, ROS, ADASYN, and SMOTE-ENN. In addition, we compared the performance of our models on imbalanced data and different balanced datasets to explore the efficiency of using balanced data on model performance.

5) FEATURE SCALING

It is important to scale the input features of the dataset within a small range. This step is important as it accelerates the learning process of the algorithm [40]. We applied one of the most common techniques used in feature scaling, which is a standard scaler. We applied this technique to each input feature in the dataset, which is one of the reasons for our approach to achieve higher accuracy. To standardize each feature in the input variables, we calculated the mean and the standard deviation for that feature. Then, the new value of X_{scaled} for each sample X is calculated as follows:

$$X_{scaled} = \frac{x-\mu}{\sigma} \tag{2}$$

where μ and σ are the mean and standard deviation of a particular feature, respectively.

C. MODEL VALIDATION

In this study, we used two common approaches of cross-validation. We describe them in the following subsections.

1) RANDOM HOLD-OUT METHOD

In this approach, we randomly divided our dataset into 80% for training and 20% for testing.

2) K-FOLD CROSS-VALIDATION

It is an improved method of cross-validation approaches used to enhance the performance of machine learning techniques, as the entire dataset is used for training and testing.

It is also a useful tool when the size of the dataset is very small, as in our study. This technique randomly divides the dataset into K subsets of equal sizes. One subset was used for the testing set, and the remaining subsets were used for the training set to build the model. This process is repeated sometimes, and the final result of the model is obtained from the average result of the testing set [41].

In our study, we used a version of k-fold cross-validation called stratified K-fold. It is used with classification problems because the class distribution in each fold is made using the same number of samples for each class. We used a stratified 5-fold cross-validation.

D. MACHINE LEARNING MODELS

We have applied some machine learning techniques, namely deep artificial neural network (DNN), decision tree (DT), logistic regression (LR), support vector classifier (SVC), K-nearest neighbor (KNN), random forest (RF), and gradient boosting (GB).

The configuration parameters of the ML methods are presented in Table 4.

TABLE 4. Machine learning methods with their parameters' settings.

ML Methods	Parameters
K-Nearest Neighbor	N_Neighbors=5, weight function=distance, leaf_size=30
Support Vector Classifier	C=1.8, Kernel=rbf, Gamma=scale, degree=1
Logistic Regression	C=1, penalty=L2, solver=sag
Decision Tree	Criterion=entropy, Splitter=best
Random Forest	N_estimators=30, Criterion='entropy', min_samples_leaf=5
Gradient Boosting	N_estimators=100, loss='deviance', learning_rate=0.1

Deep Artificial Neural Network (Deep ANN): It is one of the most common machine learning techniques because it performs similar functions to the human brain. It is a powerful tool used for modeling in the real world to solve the problem of nonlinear functions. This algorithm consists of layers, and each layer is based on processing units called artificial neurons. There is a connection between the layers and their neurons through weighted links [42], [43]. The main advantage of using a deep ANN is that it facilitates generalization and enables the network to correctly discover hidden patterns and useful knowledge from the dataset [44].

In this study, we applied a DNN to predict students' academic performance. One input layer, four hidden layers, and one output layer were considered as the architecture of the proposed DNN model.

The configuration parameters of the proposed DNN model are presented in Table 5.

TABLE 5. Parameters' configuration of DNN model.

Number of Layers	6
Layer L_1 (Input Layer)	
Number of Units	12
Layer L_2 (1st Hidden Layer)	
Number of Units	200
Activation Function	ReLU
Layer L_3 (2nd Hidden Layer)	
Number of Units	200
Activation Function	ReLU
Layer L_4 (3rd Hidden Layer)	
Number of Units	150
Activation Function	ReLU
Layer L_5 (4th Hidden Layer)	
Number of Units	100
Activation Function	ReLU
Layer L_6 (Output Layer)	
Number of Units	1
Activation Function	Sigmoid
Number of Epochs	100
Batch Size	128
Optimization Algorithm	Adam

We used 12 neurons in the input layer to represent our input features and only one neuron for the output layer, as we have a binary classification problem. We used two activation functions, namely, the rectified linear unit (ReLU) for the hidden layers and the sigmoid function for the output layer. Binary cross-entropy was used as a loss function, and the optimization algorithm Adam, which refers to adaptive moment estimation, was used to compute the errors. We used a batch size of 128 and the number of epochs used was 100.

E. EVALUATION MEASURES

In our experiments, we used five evaluation measures, namely, accuracy, precision, recall, F1-score, and classification error, to evaluate the performance of the models. Accuracy is one of the most common performance measures used in several previous studies. If the dataset has the same number of instances per class, accuracy can be a helpful measure in this case. If not, accuracy cannot be a helpful measure because the model predicts the value of the majority class. The F1-score includes indispensable and vital results regarding the performance of classifiers in each class, so it is considered as the average value of recall and precision, and it is very useful in the case of different class distributions [19], [45]. The students' instances were classified into four groups: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The evaluation measures used in our study are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F1 - Score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \tag{6}$$

$$ClassificationError = \frac{FP + FN}{TP + TN + FP + FN} \tag{7}$$

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. ENVIRONMENT

We ran our experiments on a PC with a Core i7 processor and 16 GB of RAM. We used Anaconda software (Spyder) to evaluate our proposed predictive models. Furthermore, we used two techniques for model validation: random hold-out and stratified 5-fold cross-validation to divide our dataset into training and testing.

B. RESULTS AND DISCUSSION

1) RESULTS OF THE RANDOM HOLD-OUT METHOD (80% TRAINING AND 20% TESTING)

a: PERFORMANCE EVALUATION OF THE CLASSIFIERS ON IMBALANCED DATA

Table 6 summarizes the performance of the algorithms applied in our study on an imbalanced dataset.

We have used some evaluation methods such as accuracy, precision, recall, F1-score, time, and classification error for a better understanding of the models' performance.

As mentioned above, accuracy is considered not useful in the case of an imbalanced dataset, and its results are not trustworthy. Therefore, we depended on the F1-score because this metric is better in this case.

TABLE 6. Performance evaluation of the algorithms based on the random hold-out method on imbalanced data.

Model	Test set Accuracy	F1-Score	
		Pass	Fail
DNN	89	94	28
KNN	91	95	25
SVC	91	95	0
LR	91	95	0
DT	87	93	21
RF	91	95	0
GB	91	95	9

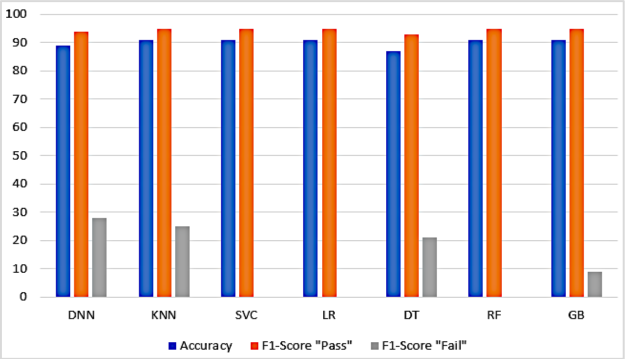

FIGURE 3. Comparison of the models' performance on the imbalanced data.

From Table 6, it is observed that the best accuracy is for KNN, SVC, LR, RF, and GB with an accuracy of 91%. The DT had the worst result, with an accuracy of 87%. In addition, the DNN achieved an accuracy of 89%.

It is observed that the DNN did not achieve the best accuracy among the others, but according to the results of the F1-score, it is the best classifier. As mentioned above, the F1-score measure includes indispensable and vital results regarding the performance of the classifiers in each class and is very useful in the case of the class imbalance problem.

As stated, we have a highly imbalanced dataset, and the distribution of the class label is not balanced. The results of the F1-score for each class show that most of the classifiers do not perform well with the "Fail" class. Therefore, solving the problem of imbalanced data is necessary to achieve better results.

For example, SVC, LR, RF, and GB fail to predict the "Fail" class; however, they have the best accuracy among the others. They achieved the highest accuracy because they only predicted the majority classes for all the predictions and did not predict the minority classes. It is observed that the results obtained from the imbalanced dataset are not acceptable because these results are unreliable. As shown in Figure 3, according to the results of the F1-score, the DNN is the best classifier among the others.

b: ACCURACY RESULTS OF THE CLASSIFIERS ON DIFFERENT BALANCED DATASETS

As mentioned above, an imbalanced dataset is a severe challenge that leads to poor performance. Therefore, solving this problem is one of the most important factors for improving the performance of the models and achieving accurate and reliable results.

TABLE 7. The accuracy of the algorithms applied on different balanced data using various resampling methods.

Model	Unbalanced Data	SMOTE	ADASYN	ROS	SMOTE ENN
DNN	89	89	88	88	81
KNN	91	75	74	77	63
SVC	91	76	73	71	67
LR	91	74	70	76	70
DT	87	84	83	86	79
RF	91	88	87	82	80
GB	91	87	87	74	79

Table 7 presents a comparison of the accuracies of the classifiers applied to different balanced datasets using various resampling methods.

Now, the classifiers consider both classes (pass and fail) after handling the problem of the imbalanced dataset, so most of them have achieved lower accuracy on different balanced datasets.

For example, the KNN achieved an accuracy of 91% using an imbalanced dataset, while this result decreased to a range of 63%–77% using different balanced datasets. The SVC achieved an accuracy of 91% using an imbalanced dataset, while this result decreased to a range of 67 %–76% using different balanced datasets. The LR achieved an accuracy of 91% using an imbalanced dataset, while this result decreased to a range of 70 %–76% using different balanced datasets. The DT achieved an accuracy of 87% using an imbalanced dataset, while this result decreased to a range of 79 %–86% using different balanced datasets. The RF achieved an accuracy of 91% using an imbalanced dataset, while this result decreased to a range of 80%–88% using different balanced datasets. The GB achieved an accuracy of 91% using an imbalanced dataset, while this result decreased to a range of 74 %–87% using different balanced datasets. In addition, the DNN achieved an accuracy of 89% using an imbalanced dataset, while this result decreased to a range of 81 %–89% using different balanced datasets.

As shown in Figure 4, the DNN is the best classifier among the others in all the balanced datasets with an accuracy of 81 %–89%.

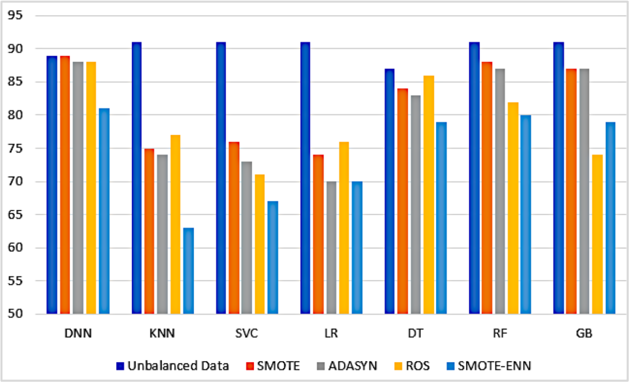

FIGURE 4. Accuracy rate of the classifiers on different balanced datasets.

It is observed that after handling the imbalanced dataset problem using various resampling methods, the obtained results are acceptable and trustworthy.

The DNN achieved an accuracy of 89% using SMOTE as an oversampling method, which is the highest result among the others, as shown in Table 7.

c: TIME AND CLASSIFICATION ERROR OF THE CLASSIFIERS

Table 8 presents a comparison of the error and time of the classifiers applied to different balanced datasets using various resampling methods.

It is observed that the DNN has achieved the lowest classification error among the others, which is equal to 0.11, using SMOTE as an oversampling method. In addition, it was observed that DNN achieved the lowest result in all the balanced data with a range of 0.11-0.19.

TABLE 8. Classification error and time of the classifiers on the different balanced datasets.

Model	Error and Time (Secs)	SMOTE	ADASYN	ROS	SMOTE ENN
DNN	Error Time	0.11 0.06	0.13 0.07	0.12 0.08	0.19 0.06
KNN	Error Time	0.25 0.07	0.26 0.07	0.23 0.08	0.37 0.07
SVC	Error Time	0.24 0.03	0.27 0.04	0.29 0.04	0.33 0.01
LR	Error Time	0.26 0.00	0.30 0.00	0.24 0.00	0.30 0.00
DT	Error Time	0.16 0.00	0.17 0.00	0.14 0.00	0.30 0.00
RF	Error Time	0.12 0.00	0.13 0.00	0.18 0.01	0.20 0.00
GB	Error Time	0.13 0.00	0.13 0.01	0.24 0.00	0.21 0.00

d: PRECISION AND RECALL RESULTS OF THE CLASSIFIERS ON DIFFERENT BALANCED DATASETS

Table 9 presents a comparison of the results of the precision and recall tests of the classifiers applied on different balanced datasets. We used the weighted average function to calculate the precision and recall for all classes.

TABLE 9. Precision and recall results of the algorithms applied on different balanced data using various resampling methods.

Model	Evaluation Metrics	Unbalanced Data	SMOTE	ADASYN	ROS	SMOTE-ENN
DNN	Precision	88	89	89	89	89
	Recall	89	89	88	88	81
KNN	Precision	88	89	89	88	88
	Recall	91	75	74	77	63
SVC	Precision	83	90	90	89	90
	Recall	91	76	73	71	67
LR	Precision	83	88	88	88	89
	Recall	91	74	70	76	70
DT	Precision	86	86	86	86	87
	Recall	87	84	83	86	79
RF	Precision	83	87	86	89	88
	Recall	91	88	87	82	80
GB	Precision	87	87	87	90	88
	Recall	91	87	87	74	79

It is observed that the recall test results are the same as the results of the accuracy described above, but the precision results of some classifiers are increased with a balanced dataset.

Regarding the precision results, it is observed that the highest result belongs to DNN and KNN with 88%, and the lowest result belongs to SVC, LR, and RF with 83% because they failed to predict the "Fail" class.

From Table 9, there are remarkable improvements in the results of some ML models. For example, the SVC achieved a precision of 83% using an imbalanced dataset, while this result was increased to 90% with different balanced datasets using SMOTE, ADASYN, and SMOTE-EEN resampling methods. The LR achieved a precision of 83% using an imbalanced dataset, whereas this result was increased to 89% with a balanced dataset using SMOTE-EEN as a hybrid resampling method. The RF achieved a precision of 83% using an imbalanced dataset, whereas this result was increased to 89% with a balanced dataset using ROS as an oversampling method. The GB achieved a precision of 87% using an imbalanced dataset, whereas this result was increased to 90% with a balanced dataset using ROS as an oversampling method. In addition, the DNN achieved a precision of 88% using an imbalanced dataset, while this result was increased to 89% with all the balanced datasets.

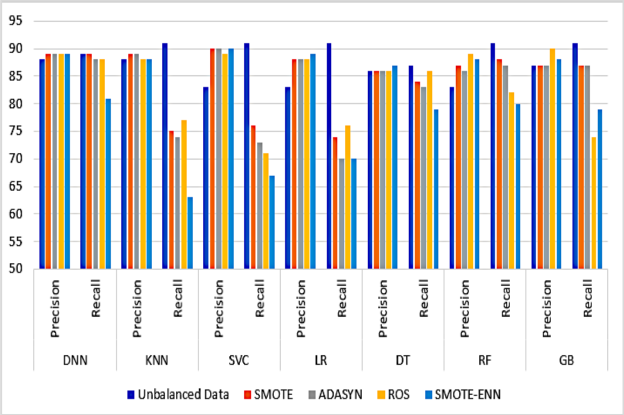

FIGURE 5. Precision and recall rate of the classifiers on different balanced datasets.

It is observed that there are slight differences in the precision results between all the applied classifiers when using different balanced datasets. All classifiers achieved a result of 86 %–90%. In addition, the SVC outperforms all other models in terms of the precision rate using different balanced datasets with a result of 89 %–90%. This result in comparison to the result obtained by the DNN model is acceptable because the DNN achieved a result of 89% with all the balanced datasets. As shown in Figure 5, the SVC is the best classifier among the others, with a precision of 90% using the SMOTE, ADASYN, and SMOTE-EEN methods.

From Table 9, regarding the recall results, the achieved results of classifiers are decreased while using different balanced datasets because the classifiers now consider all classes and do not ignore any one of them. These results are the same as those of the accuracy described above. As shown in Figure 5, it is observed that the DNN outperforms all other models in terms of the recall rate using different balanced datasets with a result of 81 %–89%.

e: F1-SCORE RESULTS OF THE CLASSIFIERS ON DIFFERENT BALANCED DATASETS

To better understand and analyze the precision and recall tests, it is more useful to use the F1-score measure.

Table 10 presents a comparison of the results of the F1-score with each class of classifiers applied on different balanced datasets using various resampling methods.

As stated above, the results of the F1-score for each class show that most of the classifiers do not perform well with all classes. Therefore, handling the problem of imbalanced data is necessary to achieve better and more accurate results. After handling this problem using various resampling methods, the results of the F1-score show that the classifiers performed well with all classes and did not ignore any one of them.

For example, the SVC, LR, and RF models ignored one of the classes while using an imbalanced dataset. Now, these models consider all classes after solving the imbalanced dataset problem using various resampling methods.

TABLE 10. F1-score results of the algorithms based on the random hold-out method on balanced data.

Model	F1-Score	Unbalanced Data	SMOTE	ADASYN	ROS	SMOTE-ENN
DNN	Average	88	89	88	88	85
	Pass	94	94	93	93	89
	Fail	28	40	37	38	36
KNN	Average	89	80	79	82	71
	Pass	95	85	84	87	75
	Fail	25	33	32	30	25
SVC	Average	87	80	79	77	74
	Pass	95	85	83	82	78
	Fail	0	34	33	30	29
LR	Average	87	79	77	81	77
	Pass	95	84	81	86	81
	Fail	0	29	27	30	28
DT	Average	87	85	85	86	82
	Pass	93	91	91	92	88
	Fail	21	22	21	23	28
RF	Average	87	87	87	85	84
	Pass	95	94	93	89	88
	Fail	0	24	22	35	34
GB	Average	88	87	87	79	85
	Pass	95	93	93	84	89
	Fail	9	28	28	33	34

As shown in Table 10, the SVC ignored the class "Fail" and failed to predict it using an imbalanced dataset, while this result was improved and increased to 34% with a balanced dataset using the SMOTE technique. The LR ignored the class "Fail" and failed to predict it using an imbalanced dataset, while this result was improved and increased to 30% with a balanced dataset using the ROS technique. The RF ignored the class "Fail" and failed to predict it using an imbalanced dataset, while this result was improved and increased to 35% with a balanced dataset using the ROS technique. The GB predicted the class "Fail" with a result of 9% using an imbalanced dataset, while this result was improved and increased to 34% with a balanced dataset using the SMOTE-ENN technique. The KNN predicted the class "Fail" with a result of 25% using an imbalanced dataset, while this result is improved and increased to 33% with a balanced dataset using the SMOTE method. The DT predicted the class "Fail" with a result of 21%, while this result is improved and increased to 28% with a balanced dataset using the SMOTE-ENN method. In addition, the DNN predicted the class "Fail" with a result of 28% using an imbalanced dataset, while this result is improved and increased to 40% with a balanced dataset using the SMOTE method.

Also, the results of the F1-score of the classifiers applied on different balanced datasets are presented in Table 10. We used the weighted average function to calculate the F1-score for all classes.

From Table 10, the KNN achieved a result of 89% using an imbalanced dataset, while this result was decreased to a range of 71 %–82% using different balanced datasets. The SVC achieved a result of 87% using an imbalanced dataset, while this result decreased to a range of 74 %–80% using different balanced datasets. The LR achieved a result of 87% using an imbalanced dataset, while this result decreased to a range of 77%–81% using different balanced datasets. The DT achieved a result of 87% using an imbalanced dataset, while this result decreased to a range of 82 %–86% using different balanced datasets. The RF achieved a result of 87% using an imbalanced dataset, while this result was decreased to a range of 84 %–85% using different balanced datasets. The GB achieved a result of 88% using an imbalanced dataset, while this result was decreased to a range of 79%–85% using different balanced datasets. In addition, the DNN achieved a result of 88% using an imbalanced dataset, while this result was decreased to 85% using the SMOTE-ENN method and increased to 89% using the SMOTE method.

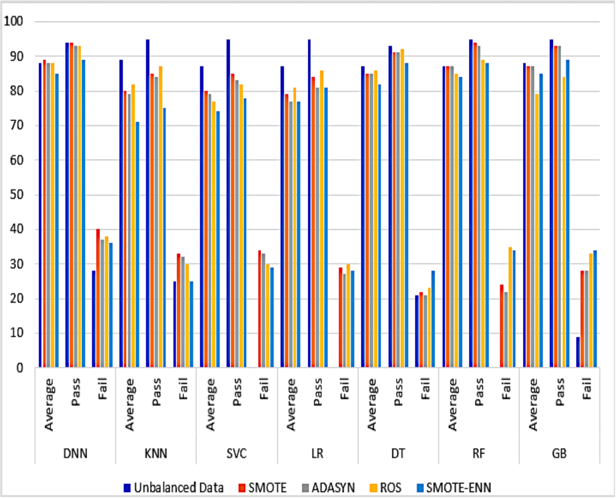

FIGURE 6. F1-score of the classifiers on different balanced datasets.

As shown in Figure 6, the DNN achieved a result of 89% using SMOTE as an oversampling method, which is the highest result among the others. However, ROS frequently achieved the highest results among all the resampling methods employed.

These results show that ROS is the most suitable oversampling method in this study. In addition, it was observed that DNN achieved the best result in all the balanced datasets, with a result of 85 %–89%.

2) RESULTS OF THE STRATIFIED 5-FOLD CROSS-VALIDATION METHOD

a: ACCURACY AND F1-SCORE RESULTS OF THE APPLIED CLASSIFIERS ON DIFFERENT BALANCED DATASETS

Table 11 indicates the obtained average results of the stratified 5-fold cross-validation of implementing models

TABLE 11. Accuracy and F1-score results of the stratified 5-fold cross-validation of the algorithms applied on different balanced data using various resampling techniques.

Model	Evaluation Metrics	Unbalanced Data	SMOTE	ADASYN	ROS	SMOTE-ENN
DNN	Accuracy	88	88	87	88	80
	F1-score	87	88	87	88	83
KNN	Accuracy	89	71	71	74	60
	F1-score	87	77	77	79	69
SVC	Accuracy	91	73	71	70	66
	F1-score	86	78	77	76	73
LR	Accuracy	91	73	69	75	67
	F1-score	86	78	76	80	74
DT	Accuracy	84	83	84	83	77
	F1-score	85	84	85	85	81
RF	Accuracy	91	87	86	79	80
	F1-score	87	87	86	82	81
GB	Accuracy	91	87	86	71	80
	F1-score	87	87	86	77	82

applied on different balanced datasets using accuracy and F1-score as evaluation measures.

The results of this strategy are more acceptable and trustworthy than those of the random hold-out method.

As mentioned above, accuracy is considered not useful in the case of an imbalanced dataset, and its results are not trustworthy. Therefore, we used the F1-score measure to better understand and analyze the results.

As shown in Table 11, The DNN achieved an accuracy of 88% and an f1-score of 88% while using SMOTE and ROS methods. The KNN achieved an accuracy of 74% and an f1-score of 79% while using the ROS technique. The SVC achieved an accuracy of 73% and an f1-score of 78% while using SMOTE technique. The LR achieved an accuracy of 75% and an f1-score of 80% while using the ROS technique. The DT achieved an accuracy of 84% and an f1-score of 85% while using the ADASYN technique. The RF and GB achieved an accuracy of 87% and an f1-score of 87% while using the SMOTE technique.

It was observed that the results obtained from the balanced datasets using SMOTE, and ROS are better than those of other resampling methods.

Regarding the achieved results of the classifiers using various resampling methods, it was observed that DNN achieved the best result in all the balanced datasets. It was observed that SMOTE achieved the highest result among all the resampling methods employed. These results show that SMOTE is the most suitable oversampling method in this study.

It is observed that the DNN outperformed the others with an accuracy result of 80 %–88% and an F1-score result with a result of 83 %–88% while using different balanced datasets, as shown in Figure 7.

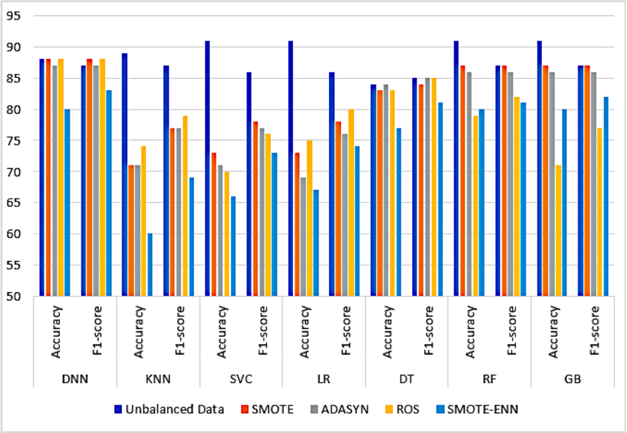

FIGURE 7. Accuracy and F1-score of the applied classifiers on different balanced datasets.

b: CLASSIFICATION ERROR OF THE CLASSIFIERS

Table 12 presents a comparison of the error and time of the classifiers applied to different balanced datasets using various resampling methods.

As shown in Table 12, the DNN achieved the lowest classification error among the others, which was equal to 0.12, using SMOTE and ROS as oversampling methods. In addition, it was observed that DNN achieved the lowest result in all the balanced data with a range of 0.12-0.20.

TABLE 12. Classification error and time of the classifiers on the different balanced datasets.

Model	Error and Time	SMOTE	ADASYN	ROS	SMOTE ENN
DNN	Error	0.12	0.13	0.12	0.20
	Time	0.14	0.13	0.18	0.16
KNN	Error	0.29	0.29	0.24	0.40
	Time	0.16	0.11	0.17	0.10
SVC	Error	0.23	0.29	0.30	0.34
	Time	0.05	0.08	0.07	0.03
LR	Error	0.23	0.31	0.25	0.33
	Time	0.00	0.00	0.00	0.00
DT	Error	0.17	0.16	0.17	0.23
	Time	0.00	0.00	0.00	0.00
RF	Error	0.13	0.14	0.21	0.20
	Time	0.00	0.00	0.00	0.00
GB	Error	0.13	0.14	0.29	0.20
	Time	0.00	0.00	0.00	0.00

3) STATISTICAL HYPOTHESIS TEST RESULTS

We applied non-parametric tests in our study since it does not make assumptions about the population distribution of the data. We used the Friedman Aligned Ranks non-parametric test since it is usually applied with a small number of algorithms in the comparison [46] and the Finner post-hoc test as it is more powerful than other tests [47] to detect the differences of all ML methods.

The achieved results of the examined methods based on F1-score values have been used to compare the performance of the ML algorithms using the above statistical tests.

a: FRIEDMAN ALIGNED RANKS NON-PARAMETRIC TEST

The null hypothesis of the Friedman Aligned Ranks non-parametric test is that the means of the results of the algorithms are the same with a significance level (alpha) of 0.05.

The results of the Friedman Aligned Ranks test are presented in Table 13 based on model validation methods used, which are random hold-out and stratified 5-fold cross-validation.

According to the results of the Friedman Aligned Ranks test, the null hypothesis that the means of F1-score values of the algorithms are the same is rejected (p-value <0.05), while the ML algorithms are ordered from the best performer to the worst one (lower-ranking value to highest-ranking value). It is observed that the DNN model prevails as shown in Table 13 as it gives better results.

TABLE 13. **Friedman aligned ranks test results.**

Algorithm	Friedman Ranking (Random hold-out)	Algorithm	Friedman Ranking (5-fold CV)
DNN	3.12500	DNN	2.50000
RF	9.00000	GB	9.87500
GB	9.87500	RF	10.50000
DT	12.75000	DT	11.87500
LR	21.50000	LR	20.62500
KNN	21.75000	SVC	22.37500
SVC	23.50000	KNN	23.75000

b: INNER POST-HOC TEST

The null hypothesis of the Finner post-hoc test is that the mean of the results of the control method and against each other algorithms is equal (compared in pairs).

The results of the Finner post hoc test are presented in Table 14 based on different model validation methods, using the DNN model as a control method.

When comparing the difference between the DNN model and the other traditional ML methods, it is observed that the null hypothesis that the mean of the results of the DNN and against each other algorithms is equal is rejected in some cases as described in Table 14.

4) COMPARISON WITH SOME PREVIOUS WORKS

We made a comparison with some previous studies that investigated the probability of students' success in upcoming courses based on their grades in the previous courses at an early stage of a semester. We found that the dataset used in previous studies [22]–[24] was small (range from 200 to 600 records) in comparison with our dataset (4266 records). In addition, predictive models are built using only traditional ML techniques. Therefore, we applied a deep artificial neural network and some traditional ML techniques using a real dataset and achieved an accuracy of 89%. From the experimental results, our proposed approach has achieved the best accuracy in comparison to all the other evaluated techniques.

V. CONCLUSION AND FUTURE WORK

Educational data mining is an important analytical tool for solving the problem of analyzing the huge amounts of educational data stored in educational settings for the decision-making process, predicting students' academic performance at an early stage of a semester, and discovering a hidden pattern and significant knowledge from educational data. There are some problems such as the imbalanced dataset in predicting students' academic performance, which is a serious challenge that leads to poor performance.

In our research, we used a dataset collected from a public 4-year university to develop our predictive models based on various machine learning algorithms, including deep artificial neural network, decision tree, random forest, gradient boosting, logistic regression, support vector machine, and K-nearest neighbor to predict students' academic performance in a data structure course based on their grades in the previous courses of the first academic year. We used various resampling techniques, such as SMOTE, ROS, ADASYN, and SMOTE-ENN to solve the imbalanced dataset problem to improve the performance of the models.

In this study, we attempt to show the effect of using an imbalanced dataset on the models' performance and to handle this problem using various resampling methods. We used two approaches for model validation: random hold-out and stratified 5-fold cross-validation.

We noticed the effect of using an imbalanced dataset on model performance. We found that none of the classifiers performed well. We noticed that the performance of the models was improved and achieved better results on the balanced data. Using the random hold-out method on the balanced dataset achieved better results, and DNN outperformed the others with an accuracy of 89%, F1-score of 89%, and a sensitivity of 89% while using the SMOTE method as an oversampling method.

Using the stratified 5-fold cross-validation on the balanced dataset has achieved reliable and accurate results, and the achieved results show that DNN outperformed the others with

TABLE 14. Finner post-hoc test results.

Comparison	Random hold-out			5-fold cross-validation		
	Statistic	Adjusted p-value	Result	Statistic	Adjusted p-value	Result
DNN vs SVC	3.50288	0.00276	H0 is rejected	3.41692	0.00190	H0 is rejected
DNN vs KNN	3.20202	0.00409	H0 is rejected	3.65331	0.00155	H0 is rejected
DNN vs LR	3.15904	0.00409	H0 is rejected	3.11606	0.00366	H0 is rejected
DNN vs DT	1.65473	0.14331	H0 is accepted	1.61175	0.15615	H0 is accepted
DNN vs GB	1.16046	0.28724	H0 is accepted	1.26791	0.20483	H0 is accepted
DNN vs RF	1.01003	0.31248	H0 is accepted	1.37536	0.19923	H0 is accepted

an accuracy of 88%, and an F1-score of 88% while using the SMOTE method as an oversampling method.

The results showed that the best result obtained when training the predictive model using DNN and a balanced dataset using SMOTE as an oversampling method was 89%.

Our research enabled us to develop a model to predict the likelihood of students' success in the upcoming data structure course and to identify at-risk students at an early stage of a semester based on their grades in the previous courses of the first academic year with acceptable results (accuracy of 89%). These results show that the DNN outperformed other prediction algorithms like support vector classifier, decision tree, logistic regression, random forest, gradient boosting, and K-nearest neighbor in terms of accuracy, recall, f1-score, and classification error metrics, while support vector classifier outperformed the DNN in terms of precision with a slight difference.

For the future work, we will extend our dataset by adding more semesters' data to improve the accuracy of the models. Other oversampling techniques, such as SVM-SMOTE and Borderline-SMOTE, will be used to evaluate their performance in comparison to the algorithms used in this study.

REFERENCES

[1] K. I. M. Ramaphosa, T. Zuva, and R. Kwuimi, "Educational data mining to improve learner performance in Gauteng primary schools," in *Proc. Int. Conf. Adv. Big Data, Comput. Data Commun. Syst. (icABCD)*, Aug. 2018, pp. 1–6.

[2] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming: 12 years later," *ACM Inroads*, vol. 10, no. 2, pp. 30–36, 2019.

[3] L. A. B. Macarini, C. Cechinel, M. F. B. Machado, V. F. C. Ramos, and R. Munoz, "Predicting students success in blended learning—Evaluating different interactions inside learning management systems," *Appl. Sci.*, vol. 9, no. 24, p. 5523, Dec. 2019.

[4] H. A. Mengash, "Using data mining techniques to predict student performance to support decision making in university admission systems," *IEEE Access*, vol. 8, pp. 55462–55470, 2020.

[5] B. Sekeroglu, K. Dimililer, and K. Tuncal, "Student performance prediction and classification using machine learning algorithms," in *Proc. 8th Int. Conf. Educ. Inf. Technol.*, Mar. 2019, pp. 7–11.

[6] S. Rai, K. A. Shastry, S. Pratap, S. Kishore, P. Mishra, and H. A. Sanjay, "Machine learning approach for student academic performance prediction," in *Evolution in Computational Intelligence*. Singapore: Springer, 2021, pp. 611–618.

[7] N. S. Sapare and S. M. Beelagi, "Comparison study of regression models for the prediction of post-graduation admissions using machine learning techniques," in *Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2021, pp. 822–828.

[8] O. Embarak, "Apply machine learning algorithms to predict at-risk students to admission period," in *Proc. 7th Int. Conf. Inf. Technol. Trends (ITT)*, Nov. 2020, pp. 190–195.

[9] E. Alyahyan and D. Düştegör, "Predicting academic success in higher education: Literature review and best practices," *Int. J. Educ. Technol. Higher Educ.*, vol. 17, no. 1, pp. 1–21, Dec. 2020.

[10] A. Farissi, H. M. Dahlan, and Samsuryadi, *Genetic Algorithm Based Feature Selection for Predicting Student's Academic Performance*. Cham, Switzerland: Springer, 2020, pp. 110–117.

[11] D. Buenaño-Fernández, D. Gil, and S. Luján-Mora, "Application of machine learning in predicting performance for computer engineering students: A case study," *Sustainability*, vol. 11, no. 10, p. 2833, May 2019.

[12] F. Ofori, E. Maina, and R. Gitonga, "Using machine learning algorithms to predict students' performance and improve learning outcome: A literature based review," *J. Inf. Technol.*, vol. 4, no. 1, pp. 33–55, 2020.

[13] F. Kouser, A. F. Meghji, and N. A. Mahoto, "Early detection of failure risks from students' data," in *Proc. Int. Conf. Emerg. Trends Smart Technol. (ICETST)*, Mar. 2020, pp. 1–6.

[14] D. Baneres, M. E. Rodriguez, and M. Serra, "An early feedback prediction system for learners at-risk within a first-year higher education course," *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 249–263, Apr. 2019.

[15] D. Das, A. K. Shakir, S. G. Rabbani, M. Rahman, S. M. Shaharum, S. Khatun, N. B. Fadilah, K. M. Qaiduzzaman, M. S. Islam, and M. S. Arman, *A Comparative Analysis of Four Classification Algorithms for University Students Performance Detection*. Singapore: Springer, 2020, pp. 415–424.

[16] M. R. Islam Rifat, A. Al Imran, and A. S. M. Badrudduza, "EduNet: A deep neural network approach for predicting CGPA of undergraduate students," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–6.

[17] A. P. Patil, K. Ganesan, and A. Kanavalli, "Effective deep learning model to predict Student grade point averages," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2017, pp. 1–6.

[18] M. A. Al-Barrak and M. Al-Razgan, "Predicting students final GPA using decision trees: A case study," *Int. J. Inf. Educ. Technol.*, vol. 6, no. 7, pp. 528–533, 2016.
[19] R. Ghorbani and R. Ghousi, "Comparing different resampling methods in predicting students' performance using machine learning techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020.
[20] I. Khan, A. Al Sadiri, A. R. Ahmad, and N. Jabeur, "Tracking student performance in introductory programming by means of machine learning," in *Proc. 4th MEC Int. Conf. Big Data Smart City (ICBDSC)*, Jan. 2019, pp. 1–6.
[21] A. Anzer, H. A. Tabaza, and J. Ali, "Predicting academic performance of students in UAE using data mining techniques," in *Proc. Int. Conf. Adv. Comput. Commun. Eng. (ICACCE)*, Jun. 2018, pp. 179–183.
[22] M. Tsiakmaki, G. Kostopoulos, G. Koutsonikos, C. Pierrakeas, S. Kotsiantis, and O. Ragos, "Predicting university students' grades based on previous academic achievements," in *Proc. 9th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2018, pp. 1–6.
[23] K. Al Mayahi and M. Al-Bahri, "Machine learning based predicting student academic success," in *Proc. 12th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2020, pp. 264–268.
[24] G. Badr, A. Algobail, H. Almutairi, and M. Almutery, "Predicting students' performance in university courses: A case study and tool in KSU mathematics department," *Proc. Comput. Sci.*, vol. 82, pp. 80–89, Jan. 2016.
[25] M. Akour, H. A. Sghaier, and O. A. Qasem, "The effectiveness of using deep learning algorithms in predicting students achievements," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 19, pp. 388–394, Jul. 2020.
[26] A. Farissi, H. M. Dahlan, and Samsuryadi, "Genetic algorithm based feature selection with ensemble methods for student academic performance prediction," *J. Phys., Conf. Ser.*, vol. 1500, Apr. 2020, Art. no. 012110.
[27] J. Sultana, M. Usha, and R. Farquad, "An efficient deep learning method to predict student's performance," Higher Educ. Qual. Assurance Enhancement, Tech. Rep., 2019.
[28] U. Pujianto, W. A. Prasetyo, and A. R. Taufani, "Students academic performance prediction with k-nearest neighbor and C4.5 on SMOTE-balanced data," in *Proc. 3rd Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Dec. 2020, pp. 348–353.
[29] J. Gajwani and P. Chakraborty, "Students' performance prediction using feature selection and supervised machine learning algorithms," in *Proc. Int. Conf. Innov. Comput. Commun.*, Singapore, 2021, pp. 347–354.
[30] M. Kumar, G. Mehta, N. Nayar, and A. Sharma, "EMT: Ensemble meta-based tree model for predicting student performance in academics," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, Art. no. 012062.
[31] J. Malini, "Analysis of factors affecting student performance evaluation using education dataminig technique," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, pp. 2413–2424, Apr. 2021.
[32] G. Kostopoulos, M. Tsiakmaki, S. Kotsiantis, and O. Ragos, "Deep dense neural network for early prediction of failure-prone students," in *Machine Learning Paradigms: Advances in Deep Learning-based Technological Applications*, G. A. Tsihrintzis L. C. Jain, Eds. Cham, Switzerland: Springer, 2020, pp. 291–306.
[33] A. M. Olalekan, O. S. Egwuche, and S. O. Olatunji, "Performance evaluation of machine learning techniques for prediction of graduating students in tertiary institution," in *Proc. Int. Conf. Math., Comput. Eng. Comput. Sci. (ICMCECS)*, Mar. 2020, pp. 1–7.
[34] D. Aggarwal, S. Mittal, and V. Bali, "Significance of non-academic parameters for predicting Student performance using ensemble learning techniques," *Int. J. Syst. Dyn. Appl.*, vol. 10, no. 3, pp. 38–49, Jul. 2021.
[35] Y. K. Salal, M. Hussain, and T. Paraskevi, "Student next assignment submission prediction using a machine learning approach," *Proc. Adv. Autom. II, Int. Russian Autom. Conf., (RusAutoConf)*, Sochi, Russia, vol. 729, Sep. 2020, pp. 383–393.
[36] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised leaning," *World Academy Sci., Eng. Technol., Int. J. Comput., Elect., Automat., Control Inf. Eng.*, vol. 1, pp. 4104–4109, Jan. 2007.
[37] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.
[38] S. T. Jishan, R. I. Rashu, N. Haque, and R. M. Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," *Decis. Anal.*, vol. 2, no. 1, pp. 1–25, Dec. 2015.
[39] H. Hassan, N. B. Ahmad, and S. Anuar, "Improved students' performance prediction for multi-class imbalanced problems using hybrid and ensemble approach in educational data mining," *J. Phys., Conf. Ser.*, vol. 1529, May 2020, Art. no. 052041.
[40] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Mining educational data to predict student's academic performance using ensemble methods," *Int. J. Database Theory Appl.*, vol. 9, no. 8, pp. 119–136, 2016.
[41] P. Kumari, P. K. Jain, and R. Pamula, "An efficient use of ensemble methods to predict students academic performance," in *Proc. 4th Int. Conf. Recent Adv. Inf. Technol. (RAIT)*, Mar. 2018, pp. 1–6.
[42] R. C. Raga and J. D. Raga, "Early prediction of student performance in blended learning courses using deep neural networks," in *Proc. Int. Symp. Educ. Technol. (ISET)*, Jul. 2019, pp. 39–43.
[43] E. T. Lau, L. Sun, and Q. Yang, "Modelling, prediction and classification of student academic performance using artificial neural networks," *Social Netw. Appl. Sci.*, vol. 1, no. 9, p. 982, Sep. 2019.
[44] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.
[45] E. S. Bhutto, I. F. Siddiqui, Q. A. Arain, and M. Anwar, "Predicting students' academic performance through supervised machine learning," in *Proc. Int. Conf. Inf. Sci. Commun. Technol. (ICISCT)*, Feb. 2020, pp. 1–6.
[46] J. L. Hodges and E. L. Lehmann, "Rank methods for combination of independent experiments in analysis of variance," in *Selected Works of E. L. Lehmann*, J. Rojo, Ed. Boston, MA, USA: Springer, 2012, pp. 403–418.
[47] H. Finner, "On a monotonicity problem in step-down multiple test procedures," *J. Amer. Stat. Assoc.*, vol. 88, no. 423, pp. 920–923, 1993.
[48] A. A. Fotouh, M. Seyam, and A. Nabil, "Predicting students' academic performance using machine learning techniques: A literature review," *Int. J. Bus. Intell. Data Mining*, vol. 1, no. 1, p. 1, 2022.

AYA NABIL received the B.Sc. degree from the Information System Department, Faculty of Computer and Information Sciences, Mansoura University, Mansoura, Egypt, in 2017. Since 2018, she has been a Demonstrator with the Department of Information System, Faculty of Computer and Information Sciences, Mansoura University. Her research interests include educational data mining and machine learning.

MOHAMMED SEYAM received the bachelor's degree in information systems from Mansoura University, Egypt, the master's degree from Cairo University, and the M.Sc. and Ph.D. degrees in computer science from Virginia Tech, USA. He is a Researcher and an Educator in the fields of software engineering, human–computer interaction, and computer science education. He has a graduate certificate in preparing future professoriate and is a fellow of the Graduate Academy for Teaching Excellence, Virginia Tech.

AHMED ABOU-ELFETOUH received the bachelor's degree in accounting information systems and the master's degree from Mansoura University, Egypt, and the Ph.D. degree in information systems from Suez Canal University. He is a Professor and an Educator in the fields of intelligent information systems, decision support systems, geographic information systems, and remote sensing. He holds the position of Dean with the Faculty of Computer and Information Sciences, Mansoura University.