IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 November 2023, accepted 2 January 2024, date of publication 5 January 2024,
date of current version 12 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350169

RESEARCH ARTICLE

Prediction of Students' Academic Performance in the Programming Fundamentals Course Using Long Short-Term Memory Neural Networks

LUIS VIVES[1], IVAN CABEZAS[2], (Member, IEEE), JUAN CARLOS VIVES[3], NILTON GERMAN REYES[4], JANET AQUINO[4], JOSE BAUTISTA CÓNDOR[5], AND S. FRANCISCO SEGURA ALTAMIRANO[4]

[1]Faculty of Engineering, Peruvian University of Applied Sciences, Lima 15023, Peru
[2]Department of Computing and Smart System, Universidad ICESI, Cali 760031, Colombia
[3]Professional School of Electrical Mechanical Engineering, Universidad Señor de Sipan, Pimentel 150131, Peru
[4]Universidad Nacional Pedro Ruiz Gallo, Lambayeque 01131, Peru
[5]Universidad Nacional de Trujillo, Trujillo 13001, Peru

Corresponding author: Luis Vives (pcsilviv@upc.edu.pe)

This work was supported by the Research Directorate of the Peruvian University of Applied Sciences.

ABSTRACT In recent years, there has been evidence of a growing interest on the part of universities to know in advance the academic performance of their students and allow them to establish timely strategies to avoid desertion and failure. One of the biggest challenges to predicting student performance is presented in the course "Programming Fundamentals" of Computer Science, Software Engineering, and Information Systems Engineering careers in Peruvian universities for high student dropout rates. The objective of this research was to explore the efficiency of Long-Short Term Memory Networks (LSTM) in the field of Educational Data Mining (EDM) to predict the academic performance of students during the seventh, eighth, twelfth, and sixteenth weeks of the academic semester, which allowed us to identify students at risk of failing the course. This research compares several predictive models, such as Deep Neural Network (DNN), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Classifier (SVM), and K-Nearest Neighbor (KNN). A major challenge machine learning algorithms face is a class imbalance in a dataset, resulting in over-fitting to the available data and, consequently, low accuracy. We use Generative Adversarial Networks (GAN) and Synthetic Minority Over-sampling Technique (SMOTE) to balance the data needed in our proposal. From the experimental results based on accuracy, precision, recall, and F1-Score, the superiority of our model is verified concerning a better classification, with 98.3% accuracy in week 8 using LSTM-GAN, followed by DNN-GAN with 98.1% accuracy.

INDEX TERMS Educational data mining, generative adversarial networks, long-short term memory, synthetic minority over-sampling technique.

I. INTRODUCTION

The Peruvian university educational process presents the challenge of generating strategies that improve the quality of teaching to form individuals with cognitive, creative, and innovative capacities. In this sense, the need arises to analyze the student dropout rate and disapproval due to economic, social, and cognitive factors. According to [1], the dropout rate of university students according to the geographical area in 2018 was: Lima-Center (13.4%), mountain (18.2%), jungle (24.6%), and coastal (24%). However, the COVID-19 pandemic increased the dropout rate, reaching 42.6% in coastal, mountain, and jungle areas and 18.1% in Lima-Center. In Engineering careers, which include the careers of Software Engineering (SE), Computer Science (CS), and Information Systems Engineering (IS) the dropout rate is between 15% to 20% [2], and the failure rates range from 25% to 30% [3]. However, the failure rate increases in the first

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

course of the specialty and ranges from 25% to 35% [4], [5]. Therefore, early identification of students at risk of dropping out or failing a course involves the analysis of attributes, characteristics, or factors during the academic process that influence academic performance.

In recent years, Educational Data Mining (EDM) has gained importance due to its results in predicting academic performance, dropout, course approval, or failure [6].

EDM is a sub-area of data mining that applies statistics and machine learning to extract, process, interpret, and evaluate hidden patterns in educational datasets [7]. Education experts utilize EDM to support academic decisions that benefit students and the academic community [8].

EDM is combined with machine learning techniques, such as Random Forest, Decision Tree, Support Vector Classifier, Logistic Regression, K-Nearest Neighbor, Deep Artificial Neural Network, and Convolutional Neural Network (CNN) whose aim is to generate predictive models based on the extraction of patterns from educational data [6], [9], [10], [11], to predict academic performance, dropout, approval, disapproval of students at an early stage. Therefore, based on the results, universities can apply strategies that reinforce student knowledge and reduce a course's dropout or failure rate [12].

One of the challenges of EDM is the amount of available data and the imbalance of data used as input in the proposed models, which causes instability in the accuracy of the results. Problems associated with unbalanced class datasets cause machine learning algorithms to converge slowly when trained, generalize overfitting to available data, and poorly resolve unseen data [13], [14].

In this research, we compare the Generative Adversarial Network [15] and the Synthetic Minority Over-sampling Technique [16] as resampling techniques to address the problem of unbalanced data and generate reliability in the results.

This study's data collected over three years belong to students from two Peruvian universities' Software Engineering, Computer Science, and Information Systems Engineering programs. A predictive model based on Long-Short Term Memory Networks was developed and compared with six models: Deep Neural Network, Decision Tree, Random Forest, Logistic Regression, Support Vector Classifier, and K-Nearest Neighbor. The proposal's framework is divided into 5 phases: Data Collection, Data Balancing, Training Data, Testing Data, and Model Evaluation. This study makes the following contributions:

- Collect and preprocess open academic data, making it available for future research.
- Evaluates two data oversampling techniques, GAN and SMOTE, for tackling the unbalanced data problem.
- Performs experimental evaluation and analysis of machine learning techniques such as Long short-term memory, Random Forest, Decision Tree, Support Vector Classifier, Logistic Regression, K-Nearest Neighbor, and Deep Artificial Neural Network.
- Evaluates quantitative performance-based accuracy, precision, recall, F1 Score, classification error, sensitivity, specificity, and confusion matrix.

This document is organized as follows: Section II discusses related work. Section III introduces the proposed method. Section IV provides an analysis of the experimental results and discusses their implications. Finally, Section V concludes the paper and outlines future work based on the results obtained.

II. RELATED WORK

In this section, we explore and analyze EDM research, focusing on applying various Machine Learning techniques. Our analysis includes performance measures, feature patterns, objectives, and algorithms used in these studies.

Recent years have seen a surge in Educational Data Mining and Machine Learning research. In their survey, [10] specifically examines the application of Artificial Neural Network techniques in EDM for predicting students' academic performance. This survey identified 21 articles, categorizing them based on objectives, education levels, predictor and output variables, algorithms, model accuracy, and key findings. They conclude that ANNs obtain accuracies above 84%. On the other hand, in [9], a systematic mapping of machine learning techniques was applied to EDM. They analyzed 39 articles and concluded that ANNs are the most used, followed by SVM, LR, and KNN.

Table 1 presents an evaluation of articles that utilize machine learning algorithms in EDM for predicting academic performance. Among the most commonly used algorithms to predict academic performance are ANN [17], [20], [21], [23], [24], [25], [27], [30], [32], [33], LR [18], [26], [27], [29], [30], [32], DT [18], [21], [22], [23], [24], [25], [29], RF [21], [24], [25], [26], [27], [29], SVC [22], [25], [26], [27], [29], [30], LSTM [24], [25], [30], [31], [33], KNN [26], [27], [29], K-means [28], [34], [35], DNN [19], [29], NB [22], [26], Bagging [21], [22], Boosting [17], [21], CNN [22], and GB [29]. In [34], 47 predictive models were evaluated. We identified that traditional algorithms, such as ANN, LR, RF, KNN, SVC, and K-means, have been widely used in EDM, while the current trend is to rely on predictive algorithms such as LSTM, DNN, and CNN.

Moreover, we can observe the limited use of data-balancing algorithms in research. In [29], four data balancing algorithms were compared - SMOTE, ADASYN, ROS, SMOTE-ENN -, to handle unbalanced data sets and improve GB, LR, SVC, and KNN. The authors determined that the Synthetic Minority Over-sampling Technique (SMOTE) yields superior results in managing unbalanced datasets. In this context, Deep Neural Networks (DNNs) achieved an accuracy of 89%, closely followed by Random Forests at 88%. We infer that data balancing not only facilitates achieving class equilibrium but also mitigates the bias associated with class disproportionality. Furthermore, it provides a more substantial dataset for training, thereby positively influencing the enhancement of performance metrics.

TABLE 1. Summary of review of some previous work on student academic performance using machine learning.

Reference	Year	Country of study	Propose	Algorithm models used	Dataset	Number of attributes	Performance Metrics	Highest accuracy & Best Algorithm	Data Balanced
[17]	2021	Portugal	ANN application to predict student performance centered on economic environment data	ANN Boosting Bagging	649	33	Accuracy True Positive Rate False Positive rate Precision Recall F-Measure Confusion Matrix	88% (Bagging)	No
[18]	2021	United States	Application of ML to predict concepts/skills to write computer programs	LR DT	145	4	Accuracy precision Recall F1-Score AUC	84% (DT)	No
[19]	2020	Greece	Application of transfer learning with DNN to predict student performance.	Transfer learning with DNN	192	-	Accuracy T-test	86%	No
[20]	2020	Ecuador	ANN application to predict student performance with academic and socioeconomic data.	ANN	1308	5	Accuracy Precision Recall	74.5% (ANN)	No
[21]	2020	-	Application of machine learning to predict dropout	DT ANN RF Voting Bagging boosting	480	16	Geometric Mean Precision True Negative Rate True Positive Rate F1-Score Area Under Curve	85% (GA+RF)	No
[22]	2020	India	CNN application to predict if a student can complete the course.	CNN NB DT SVC	480	16	Accuracy Recall Precision F-measures	90% (CNN)	No
[23]	2020	Cuban	Application of machine learning to predict dropout	DT ANN	456	19	Accuracy precision Recall F-Measure	96.71% (ANN, DT)	No
[24]	2023	Yemen	Using ANN-LSTM for multi-class classification.	ANN+LSTM RNN GRU DFFNN RF AML	32593 students 7 courses	206	Accuracy Precision Recall MAP MAR	ANN+LSTM 68.8% (Week 15) 71.35% (Week 25)	No
[25]	2023	India	Using LSTM with random forest and gradient boosting with a 4-layer architecture to predict student performance.	LSTM+RF+GB RNN CNN ANN LSTM NB DT RF SVM	32593	30	Accuracy Precision Recall F-measure	LSTM+RF+GB 96.40%	No
[26]	2022	Turkish	ML application to predict the final grade.	RF KNN SVC LR NB	1854	3	Confusion matrix Accuracy Precision Recall F-Score AUC	74.6% (RF, NN)	No
[27]	2022	Saudi Arabia	ML application to predict academic performance.	SVC RF KNN ANN LR	842	10	Mean Absolute Error (MAE) Mean Absolute Percentage Error (MAPE	93.7% (RF)	No

TABLE 1. *(Continued.)* **Summary of review of some previous work on student academic performance using machine learning.**

[28]	2022	Japan	Proposes framework using unsupervised algorithms (k-means)	K-means	537	70 000 real-world problem-solving	Without information	Without information	No
[29]	2021	Egypt	ML application to predict disapprovals and dropouts.	DNN DT RF GB LR SVC KNN	4266	12	Accuracy precision Recall F1-Score classification error	Using SMOTE for data imbalance handling 89% (DNN) 88% (RF) 87% (GB) 84% (DT) 76% (SVC) 75% (KNN) 74% (LR)	SMOTE ADASYN ROS SMOTE-ENN
[30]	2019	Saudi Arabia	Using LSTM to predict student performance by weeks	LSTM ANN SVM LR	23593	20	Accuracy Recall	LSTM achieved an accuracy of 93.46% in week 38.	No
[31]	2019	China	Application of CNN, LSTM, and SVM to predict student performance.	(CNN-LSTM-SVC) (CNN-LSTM) (CNN-RNN)	39 courses	11	Precision Recall F1-Score AUC	(CNN-LSTM-SVC 91.55%	No
[32]	2019	-	LSTM application to predict the withdrawal of courses	LSTM ANN LR	25 541	20	Accuracy Loss Precision Recall	LSTM (97.25 % in 25 weeks) LSTM (84.15% in 10 weeks)	No
[33]	2019	-	Application of machine learning to predict academic assistance for Students	ANN	900	10	Accuracy Recall	97.4% (ANN)	No
[34]	2019	Turkish	Constructing an ensemble meta-base classifier technique to predict students' performance.	47 models	400	13	Accuracy Recall F-measure ROC metric	98.5% (Naive Bayes + Adaboots_J48)	No
[35]	2018	-	ANN application to predict student performance	LR K-means	284	5	Without information	LR, K-means (50%)	No

Likewise, the amount of data and attributes used to train predictive models varies among the research. In [24] and [25], they used 32593 student records and considered 206 attributes corresponding to demographics and academic data. In [32], they used 25,541 student records collected over nine months and analyzed 20 attributes from an online platform where notes on activities such as forums, quizzes, and tests stand out. In [29], they used 4,266 records and considered 12 attributes corresponding to grades from different courses. In [26], they analyzed 1,854 records and only three attributes (previous exams, school data, and faculty data). In [20], they analyzed 1,308 records with five attributes (score, vulnerability index, regime, gender, and population segment). In [33], they used 900 records with ten attributes obtained from the interaction of students with an online platform. In [27], they used 842 records with ten attributes distributed in personal and academic attributes. In [17], they used 649 records with 33 attributes extracted from 3 categories (personal attributes, academic background, economic background). In [28], they used 537 attributes; no specific use was evident. In [21] and [22], they used 480 records with sixteen attributes distributed in three categories (demographic category, academic category, and behavioral category). In [34], they used 400 records with thirteen attributes corresponding to academic and personal data. In [35], they used 284 with five attributes extracted from the interaction of students with a virtual platform (view, post, forum view, forum post, successful submission). In [18], they used 145 records with four attributes (repetition concept, selection concept, repetition skills, and method skills). The amount of data and an adequate number of attributes are necessary for predictive models to learn from the interaction

of the data and its attributes. Predictive models such as DNN, LSTM, and CNN require much data to learn and predict correctly.

On the other hand, the accuracy percentage varies according to the predictive model used, the data set, and the trained attributes. In [34], they achieved an accuracy of 98.5%. The authors combined Naive Bayes (NB) with ababoots_J48. The authors constructed an ensemble meta-based tree model that combines a boosting method with Naive Bayes trees to predict student performance. They used the Pearson correlation method to find attributes with high correlation, with the visited resource attribute having a high impact on the final result.

Meanwhile, in [33], they achieve an accuracy of 97.4% using ANN. The authors manage to predict whether a student requires academic assistance in their course using an ANN.

According to the proposed architecture, the ANN was designed with a network of 4 input neurons, 12 hidden layers, and three outputs. Likewise, in [32], the authors used Long Short-Term Memory (LSTM) to predict course withdrawal. LSTM achieved an accuracy of 97.25% to predict a student's withdrawal in week 25 and 84.15% to predict a student's withdrawal in week 10. In [23], they used Decision Trees and Artificial Neural Networks to predict student dropout in an undergraduate program. The classification comprised two (promoted or not promoted) and three classes (promotion, repetition, dropout). They conclude that ANN and DT achieve an accuracy of 96.71% using all variables for both classes.

Meanwhile, in [25], they propose using LSTM with RF and gradient boosting with a 4-layer architecture to predict student performance. They were compared with eight predictive models and achieved the best accuracy, 95.40%. In [30], they propose using LSTM with three layers to predict for weeks, whether a student passes or fails a course. They used data from virtual learning environments and comparisons against ANN, SVM and LR. LSTM achieved an accuracy of 93.46% in predicting a student's performance in week 8, ANN achieved 85%, SVM achieved 75%, and LR achieved 80% in the same week. In [31], the researchers utilized CNN, LSTM, and SVC in their study. The CNN was employed for feature extraction, while the LSTM model was utilized to retain historical data. Additionally, SVC was employed to address the issue of data imbalance. The authors compared their predictive model against several established models, including CNN-LSTM, CNN-RNN, RF, DT, LR, and SVC. The proposed model achieves 91.55%.

On the other hand, in [22], they propose using a Convolutional Neural Network to predict whether a student will complete their course. The authors demonstrate that CNN achieves an accuracy of 90%. However, they conclude that the small amount of data and variables affect the model's accuracy. In [29], they used Deep Neural Networks to predict undergraduate students' pass and dropout rates. They compared data balancing algorithms to handle unbalanced data sets. They concluded that Deep Neural Network achieved 89% accuracy, followed by Random Forest with 88%. In [17], they trained an Artificial Neural Network to predict student performance using economic environment data. The authors compared three predictive models: ANN, Booting, and Bagging. They conclude that Bagging classifiers achieve a better accuracy of 88% and consider economic data important for predicting student performance. In [21] they used Genetic Algorithm (GA) to select features and Random Forest to classify students according to their performance. They compared six predictive models: DT, ANN, RF, Voting, Bagging, and Boosting. The authors concluded that the use of GA+RF achieves an accuracy of 85%. However, it is considered for future work to obtain more training data and compare other algorithms for feature selection. In [18] they managed to predict concepts/skills for writing computer programs using DT and LR. The results show that DT achieves 84% accuracy. The authors conclude that the concept of selective logic is an essential prerequisite for writing computer programs, and they also suggest evaluating the models with a larger amount of training data. In [19], the authors used Transfer Learning with DNN in response to the limited quantity of available data to predict student performance. An accuracy rate of 86% was attained. The prediction models in question were not evaluated by comparative analysis. In [26], they compared six models: RF, NB, NN, SVC, LR, and KNN, to predict the final grade of undergraduate students. They found that Random Forest and Nearest Neighbors achieve 74.6% accuracy while KNN achieves 69.9%. The authors suggest that other training variables should be used to improve the accuracy of the models. In [20], they trained an ANN with academic and socioeconomic data to predict students who fail in undergraduate programs. They used two predictive models. The first model presents 48 input neurons, 39 hidden layers, and one output, while the second model presents seven input neurons, four hidden layers, and one output as the network architecture. The authors achieved 74.5% accuracy with the second ANN model. However, they consider that the number of variables is limited for their training.

In [24], they used ANN-LSTM for multi-class classification (distinction, pass, fail, and withdrawn). The architecture of the LSTM network is composed of i) an input layer for 200 attributes, ii) a dense hidden layer with an output of 100 units and an activation function "ReLu", and iii) an output layer with function "SoftMax" activation with four output units representing four categories: Distinction, Pass, Fail, and Withdrawn.

In [35], the authors seek to predict academic performance in online learning systems using a linear regression model and a k-means classifier. Both algorithms achieved an accuracy of 50%. The authors recommend using a dataset with more records and attributes. Also, other algorithms should be evaluated, and performance measures of the models should

TABLE 2. Dataset attribute list.

Attribute name	Type	Values
Year of admission	Numeric	2020, 2021, 2022
Career	Nominal	Software Engineering (SE), Computer Science (CS), Information Systems Engineering (IS)
Gender	Nominal	Male(m), female(f)
Qualified Practice 1	Numeric	[0-20]
Partial Task	Numeric	[0-20]
Midterm Exam	Numeric	[0-20]
Qualified Practice 2	Numeric	[0-20]
Final Task	Numeric	[0-20]
Participation in Class	Numeric	[0-20]
Final Exam	Numeric	[0-20]
Linguistic Comprehension	Numeric	[0-20]
Mathematics	Numeric	[0-20]
Target	Nominal	[Pass, Fail]

be tested. In [28], they lack the results of their evaluations on k-means.

Based on the related works discussed, the research gap addressed in this study is described as follows. Firstly, most of the proposed models [20], [26], [27], [29], [30], [32], [33], aim to predict student performance using imbalanced data. Secondly, while many studies propose predictive models for assessing student performance at the end of a course, instructors often require weekly or even daily predictions.

In this research, seven machine learning algorithms (DNN, LSTM, DT, RF, LR, SVC, and KNN) were employed to evaluate results in terms of student performance prediction. Studies [24], [25], [28], [30], [31], [32] utilized LSTM. However, in our research, we applied and compared the prediction results using original data and synthetic data generated by two data balancing methods (SMOTE and GAN). We used 5-fold stratified cross-validation to stabilize the resulting evaluation measures.

III. RESEARCH METHOD

In this section, Fig. 1 shows the proposed approach flowchart, involving i) data collection, ii) data preprocessing, iii) data balancing, iv) training data, v) testing data, and vi) model validation:

A. DATA COLLECTION

The data collected were from university students in Software Engineering, Information Systems Engineering, and Computer Science careers from two Peruvian universities from 2020-2022. We obtained 677 records with 13 academic attributes related to academic grades in programming fundamentals, linguistic comprehension, and mathematics. Table 2 presents the academic attributes of the data set used to predict whether a student passes or fails the programming fundamentals course. Demographic and family data were not considered due to data restrictions and privacy by universities.

B. DATA PRE-PROCESSING

We carried out the process of data cleaning, data discretization, and feature encoding to obtain a unified and error-free data set, which allows for better results.

1) DATA CLEANING

The data cleaning process was carried out to eliminate records with missing data. We were able to identify six records with one or more missing attributes.

2) DATA DISCRETIZATION

The discretization process allowed us to transform the numerical values of student grades into nominal values. Since the goal is to classify whether a student passes or fails the programming fundamentals course, the label "Passed" considers the grade range from 12.5 to 20. In contrast, the label "Failed" considers a grade from 0 to 12.49, as illustrated in Table 3.

TABLE 3. Attribute discretization.

Attribute name	Value	Student marks
Target	Pass	**>=12.5**
	Fail	**<12.5**

3) FEATURE ENCODING

In the feature encoding stage, nominal data was taken to be converted into numerical labels. In Table 4, we can see the attributes, their values, and their encoded label.

TABLE 4. Feature encoding of attributes.

Attribute name	Values	Label encoding
Career	Software engineering (SE)	1
	Computer Science (CS)	2
	Information Systems (IS)	3
Gender	Male(M)	1
	Female(F)	0
Target	Pass	1
	Fail	0

C. DATA BALANCING

To solve the problem of unbalanced data, which leads to the domination of majority classes when training and testing machine learning models, we used two techniques: i) The Synthetic Minority Over-Sampling Technique (SMOTE) [16] and ii) Generative Adversarial Networks (GAN) [15], [36].

Fig. 2 presents the data imbalance on the attributes of the class label, where the majority class (Pass) represents 68% of the data and the minority class (Fail) accounts for 32% of the data.

In Fig. 3, the analysis of attribute correlation is presented. There is a 51% correlation between graded practice one and

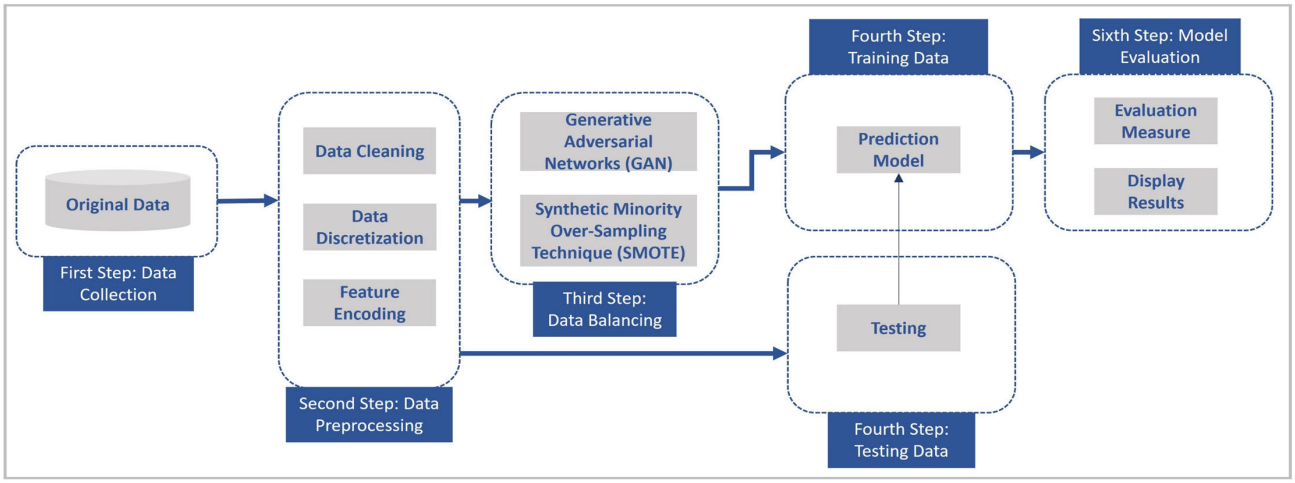


FIGURE 1. Flow diagram of the proposed approach.

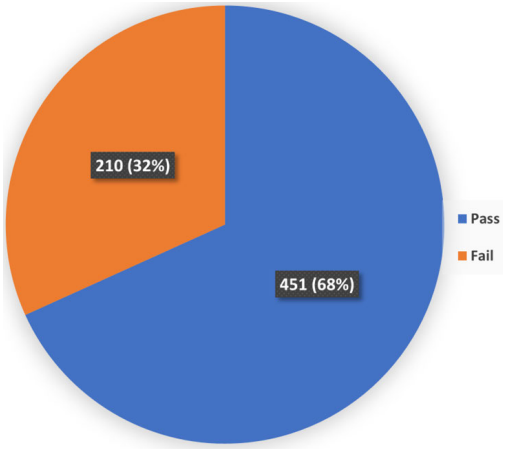


FIGURE 2. Imbalanced distribution of the class label.

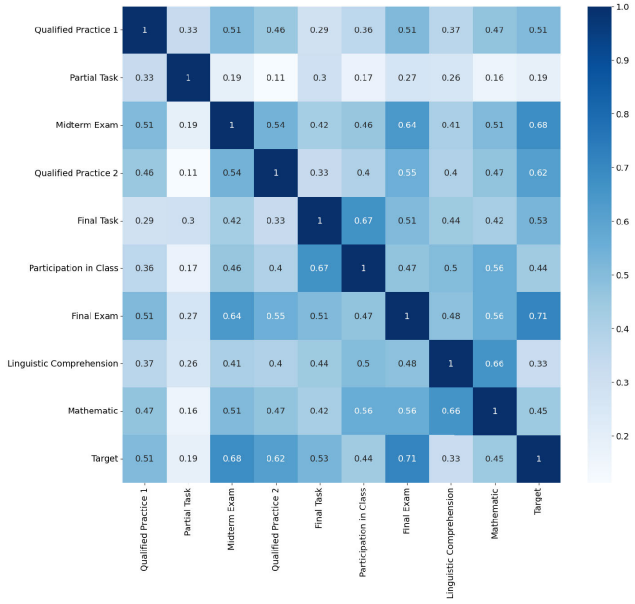


FIGURE 3. Correlation between the attributes.

the midterm exam, a 51% correlation between graded practice one and the final exam, and a 51% correlation between graded practice one and the target. Similarly, the correlation between the midterm and final exams is 64%, and between the midterm exam and the target, it is 68%. Moreover, graded practice two correlates with the final exam and the 55% and 62% target, respectively. The correlation between the final project and class participation is 67%. Finally, a correlation of 71% is observed between the final exam and the target.

D. TRAINING DATA

For the training process, we configured seven machine learning techniques: Long Short-Term Memory, Deep Neural Network, Decision Tree, Random Forest, Logistic Regression, Support Vector Machine, and K-Nearest Neighbor. Table 5 presents the configuration parameters of the machine learning models used in this research.

Long Short-Term Memory is a Recurrent Neural Network created by Hochreiter and Schmidhuber in 1997 [37] to address the problems of explosion and disappearance of gradient obtained in traditional RNN models.

LSTM has been used in time series problems [38], [39], [40].

In this study, we will use LSTM to predict student performance. The configuration parameters of the proposed model are presented in Table 6.

An LSTM network contains four main components: i) cell state, ii) input gate, iii) output gate, and iv) forget gate. The input gate, cell state, and output gate are necessary to update, maintain, and delete information from the forget gate. The architecture of an LSTM cell and its components is shown in Fig. 4.

The forget gate at a given time t(f_t) is designed using a neural network and a sigmoid function. It receives as input a data point representing the current state at time t(X_t) and the hidden state of a previous data point (h_{t-1}), concatenates them, and applies the sigmoid function, yielding a value

TABLE 5. Parameter configuration of each machine learning model.

Machine learning model	Parameters
Deep Artificial Neural Network	(64, activation='relu', input_shape=13) (32, activation='relu') (2, activation='softmax') (optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
Decision Tree	DecisionTreeClassifier (Criterion="gini", Spliter="best", random_state=42)
Random Forest	RandomForestClassifier (N_estimator=100, criterion="gini", random_state=42, n_jobs=-1)
Logistic Regression	LogisticRegression (random_state=42, n_jobs=-1, penalty='l2')
Support Vector	SVC (random_state=42, C=1.0, kernel='rbf', degree=3, gamma='scale')
K-Nearest Neighbor	KNeighborsClassifier (random_state=42, n_neighbors=5, weights='uniform', n_jobs=-1)

TABLE 6. Parameter configuration for the LSTM model.

Machine Learning model	Parameters
Long Short-Term Memory	Number of layers = 3 (64, activation='relu', input_shape=13) (32, activation='relu') (2, activation='softmax') (optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

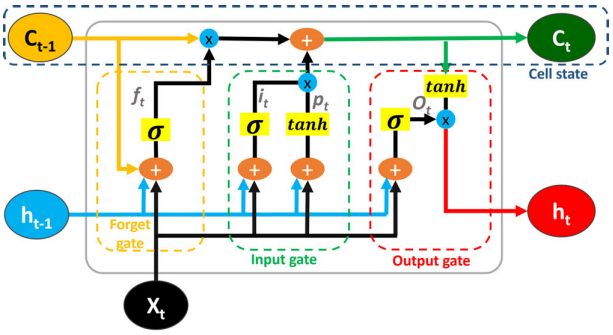


FIGURE 4. The architecture of the LSTM cell.

between 0 and 1. A value of 1 signifies that the outcome will be retained, while 0 indicates that the outcome will be discarded. This process is described in Equation 1.

$$f_t = \sigma\left(W_f\left[h_{(t-1)}X_{(t)}\right] + b_f\right) \tag{1}$$

The input gate facilitates updating the cell's current state and comprises two steps. The first step involves obtaining information (it) by multiplying the input (X_t) and the hidden state from a previous time (h_{t-1}), concatenating them, and applying the sigmoid function. The resulting value determines whether the information is retained or rejected, as described in Equation 2. The second step calculates (pt) using the same current state information (X_t) and the hidden state (ht-1)concatenating them in a tanh function, expressed in Equation 3.

$$i_t = \sigma\left(W_i\left[h_{(t-1)}, X_{(t)}\right] + b_i\right) \tag{2}$$

$$p_t = \tanh\left(W_p\left[h_{(t-1)}, X_{(t)}\right] + b_p\right) \tag{3}$$

Algorithm 1 Pseudocode of LSTM Layer

Input: $X = (X_1, X_2, \ldots, X_n)$
Output: $h = (h_1, h_2, \ldots, h_n)$
Given parameters: $W_f, W_i, W_p, W_o, b_f, b_i, b_p, b_o$
Initialize: $h_{t-1} = 0, C_{t-1} = 0$
for *each input X at time* **do**
  Calculate: Forget gate f_t = (1);
  Calculate: Input gate i_t = (2), and p_t = (3);
  Calculate: Cell state C_t = (4);
  Calculate: Output gate O_t = (5);
  Calculate: New state h_t = (6) ;
End for
Apply the activation function to obtain the output of the LSTM layer.

The cell state, or the update of the cell state, utilizes information from both the forget gate and input gate to decide and store the new state in the cell of state. The previous cell state (Ct-1) is multiplied by the vector (ft). If the result is 0, then the values are discarded. If the result is 1, then the previous memory state is completely passed to the cell, allowing the calculation of the new state by taking the output values of the vectors (it) y (pt). This process is described in Equation 4.

$$C_t = (C_{t-1} * f_t) + (i_t * p_t) \tag{4}$$

An output gate determines the value of the following hidden state (h_t). First, it multiplies the previous hidden state (h_{t-1}) with the current state (X_t) concatenated in a sigmoid function, as shown in Equation 5. Then, it updates the cell state (C_t) by multiplying it by a tanh function. Finally, it predicts the student's performance by obtaining h_t, as described in Equation 6.

$$o_t = \sigma\left(W_o\left[h_{t-1}, X_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

In Equations 1, 2, 3, 4 y 5, W represents the weights of the gates, and b represents the biases. Algorithm 1 shows the pseudocode of the LSTM layer. The input for each student's learning notes is distributed at each time t, from week 4 to week 16. It is represented in the vector $X = [X_i, \ldots. X_t]$, The data for the forget gate, input gate, cell state, and output gate are calculated, and the new state in the vector ht is obtained. An activation function is applied to obtain the output of the LSTM layer, which allows for predicting whether a student passes or fails the course.

The design of our LSTM network consists of i) an LSTM encoder with 64 neurons that receives inputs ranging from 5 to 12 attributes depending on the week being evaluated, using a ReLu activation function, ii) an LSTM Decoder with a 32-neuron LSTM layer and a ReLu function, and it also has a Dense layer with two neurons that connect all outputs from the previous layer through the

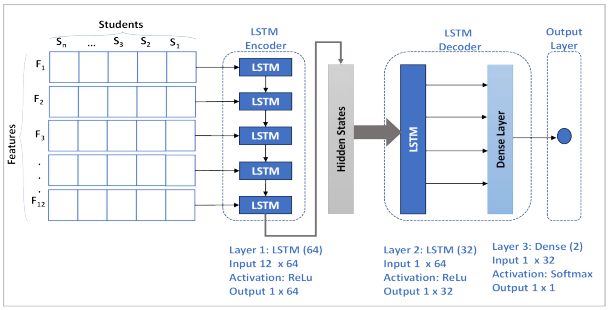


FIGURE 5. Architecture LSTM model.

SoftMax function. The output is 0, indicating that the model predicts the student will fail the course, and 1 indicates that the model predicts the student will pass the course. Fig. 5 shows the details of the LSTM architecture used in this research.

E. TESTING DATA

We applied K-fold stratified cross-validation, where k=5 for all evaluated models.

F. MODEL VALIDATION

To verify the results of our models, we used five evaluation measures: accuracy, precision, recall, F1-Score, and classification error. Accuracy measures the quality of our model 7. Precision allows us to measure the percentage of cases the model gets right 8. Recall provides us with information about the number of cases the model can identify 9. F1-Score compares performance by combining precision and recall 10. The classification error lets you know the percentage of error our model generates 11. The outcomes of our model in comparison to the existing outcomes were categorized as follows: true positive (PT), true negative (TN), false positive (FP), and false negative (FN). Sensitivity is a measure that indicates the likelihood that a student who has actually passed the test will be correctly identified as having passed by the predictive model. On the other hand, specificity indicates the likelihood that a student who has actually failed the test will be correctly identified as having failed by the predictive model.

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

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - Score = 2 * \left(\frac{Recall * Precision}{Precision + Recall} \right) \tag{10}$$

$$ClassificationError = \left(\frac{FP + FN}{TP + TN + FP + FN} \right) \tag{11}$$

$$Sensitivity = TP/(TP + FN) \tag{12}$$

$$Specificity = TN/(TN + FP) \tag{13}$$

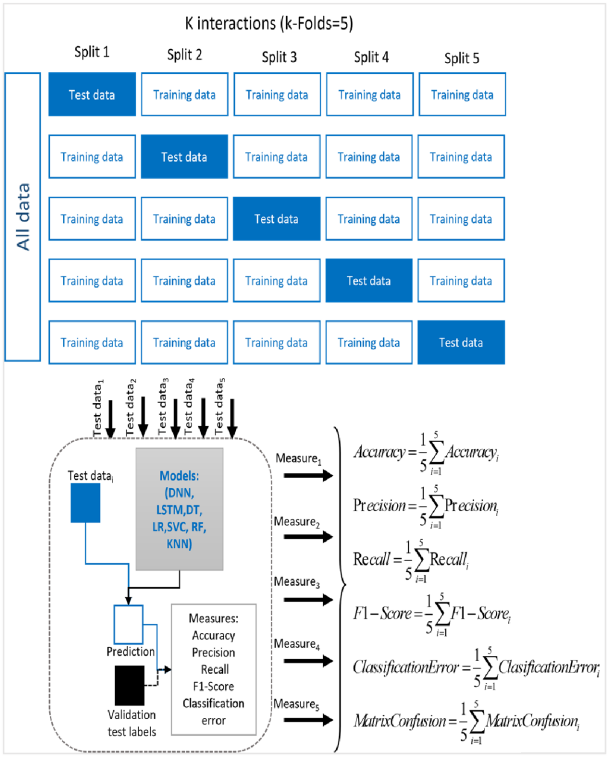


FIGURE 6. Diagram of the 5-fold cross-validation method.

K-fold stratified cross-validation was applied five times. This process ensures that the evaluation measures (accuracy, precision, recall, F1-Score, classification error, and confusion matrix) are obtained through the average of the five iterations generated by cross-validation. Fig. 6 shows the details of obtaining the evaluation measures after comparing the model's results with the test data in each iteration.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. ENVIRONMENT

This research used a laptop with a Ryzen 7-5800X processor, 16 GB of RAM, and a 1 TB hard drive. The source code was developed in Python, using Jupyter Notebook. We used Python libraries such as NumPy, Matplotlib, Pandas, Scikit-Learn, Keras, and TensorFlow.

B. RESULTS AND DISCUSSION

The experimental results were carried out to verify that the proposed technique of using LSTM and EDM to predict students' performance in the programming fundamentals course achieves high performance over the other evaluated techniques. In this sense, we evaluated i) The performance of the classifiers on the data set without applying balancing techniques, ii) The performance of the classifiers applying data balancing techniques, and iii) The performance of classifiers specifically in week 8.

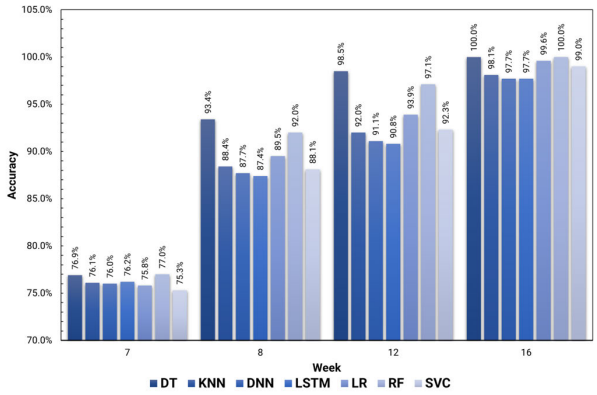


FIGURE 7. Accuracy of predictive models distributed by weeks.

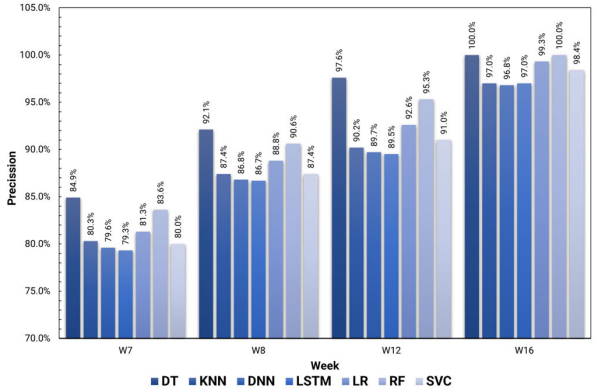


FIGURE 8. Precision of predictive models distributed by weeks.

1) PERFORMANCE EVALUATION OF THE CLASSIFIERS ON THE DATASET WITHOUT APPLYING BALANCING TECHNIQUES

Performance measures such as precision, Recall, and F1-Score were applied to the seven predictive models (Long Short-Term Memory, Deep Neural Network, Decision Tree, Random Forest, Logistic Regression, Support Vector Classifier, and K-Nearest Neighbor). The dataset with unbalanced data was used.

Fig. 7 presents the accuracy results achieved by the predictive models and their distribution by week. In week 7, RF achieved the highest accuracy of 77%, while SVC recorded the lowest at 75.3%. In weeks 8, 12, and 16, DT maintains an increase in accuracy ranging from 93.4% to 100%, followed by RF, achieving an accuracy of 92%, 97.1%, and 100% in weeks 8, 12, and 16, respectively. However, the LSTM algorithmic model reaches 76.2%, 87.4%, 90.8%, and 97.7% in weeks 7, 8, 12, and 16. This is attributed to the limited data quantity and bias due to imbalanced data, which allows traditional models to focus their predictions on the majority class (Passed).

Regarding the precision measure, in weeks 7, 8, 12, and 16, DT better classify approved cases, achieving 84.9%, 92.1%, 97.6%, and 100% accuracy, respectively. RF exhibits a similar pattern, better classifying approved student cases with percentages of 83.6%, 90.6%, 95.3%, and 100% in weeks 7, 8, 12, and 16. LSTM achieves the lowest percentage in classifying approved students in weeks 7, 8, and 12 with values of 79.3%, 86.7%, and 89.5%, and in week 16, it only surpasses DNN with a precision of 97%. Fig. 8 displays the weekly results of the recall measure for each algorithmic model.

Regarding the recall measure, in week 7, KNN achieved 12% in classifying false negatives, meaning students are classified as failed when their current status is passed. In week 8, DT achieves a recall of 97.9%, indicating that 2.1% of students are classified as failed while their actual status is passed. In week 12, DT achieved a recall of 99.7%, and in week 16, DT, LR, and RF all achieved a recall of 100%.

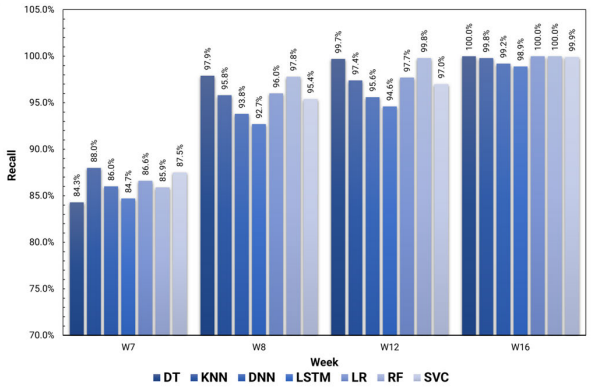


FIGURE 9. Recall of predictive models distributed by weeks.

Fig. 9 displays the weekly results of the recall measure by each algorithmic model.

Regarding the F1-Score, in week 7, KNN and SVC show better management of imbalanced data with 88% and 87.5%, respectively. Similarly, in week 8, DT and RF achieved the highest values of 97.9% and 97.8% for managing imbalanced data. In week 12, DT and RF continue to obtain the best values, and finally, in week 16, DT, LR, and RF achieve a 100% F1-Score. LSTM presents 84.7%, 92.7%, 94.6%, and 98.9% in weeks 7, 8, 12, and 16, respectively. Fig. 10 displays the weekly results of the F1-Score measure for each predictive model.

2) EVALUATION OF THE PERFORMANCE OF THE CLASSIFIERS APPLYING DATA BALANCING TECHNIQUES

The problem of unbalanced data and the small amount of data for training machine learning predictive models lead to biased results and performance. Data balancing techniques such as SMOTE and GAN were used to address both problems. Stratified cross-validation was performed five times. The synthetic data for SMOTE and GAN were 4000. However, after obtaining the data with GAN, a data cleaning was carried out, as it generated 64 inconsistent records.

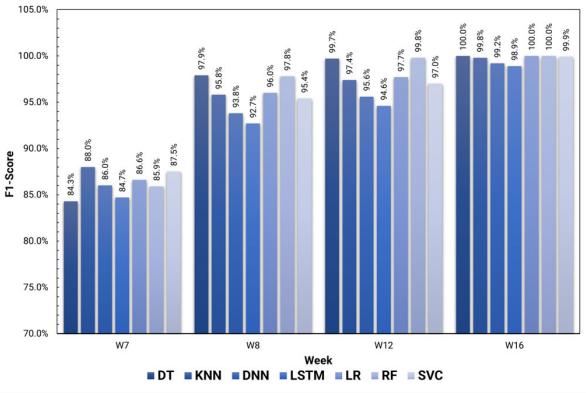


FIGURE 10. F1-Score of predictive models distributed by weeks.

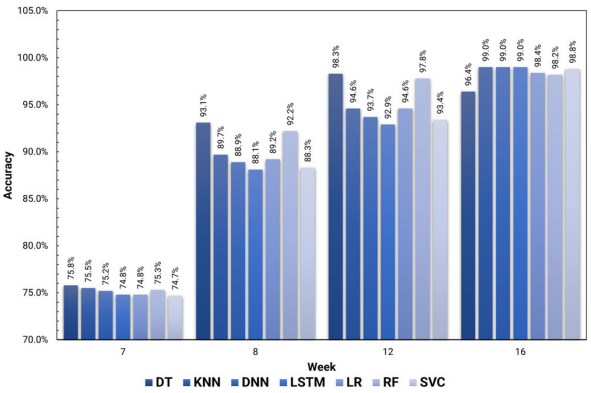


FIGURE 11. Accuracy of predictive models distributed by weeks with SMOTE.

a: PERFORMANCE EVALUATION OF THE MODELS OF MACHINE LEARNING USING THE SMOTE TECHNIQUE

Fig. 11 shows the results of the seven predictive models based on their accuracy using SMOTE to classify two categories (pass and fail). It can be observed that DT achieves a precision of 75.8%, 93.1%, and 98.3% in weeks 7, 8, and 12, respectively. However, in week 16, the DNN, LSTM, and KNN models reach a precision of 99% accuracy.

Regarding the precision measure, KNN with SMOTE in week 7 achieves the best precision measure with 83.9%, equating to a 16.1% error rate in classifying false positives, meaning the model considers students as passed, whereas they actually failed. In weeks 8 and 12, DT achieves the best precision measure with 92.1% and 98.1%, respectively. However, in week 16, KNN, DNN, and LSTM achieved better management of false positives, with a precision of 98.7%. Fig. 12 displays the results of the precision measure with SMOTE.

Regarding the recall measure, DT and SVC with SMOTE achieve better results in classifying false negatives. Fig. 13 shows the results of the recall measure with SMOTE for the predictive models distributed by week.

Regarding the F1-Score measure, DT with SMOTE emerges as the best classifier in weeks 7, 8, and 12 with percentages of 81.8%, 94.3%, and 98.8%, respectively.

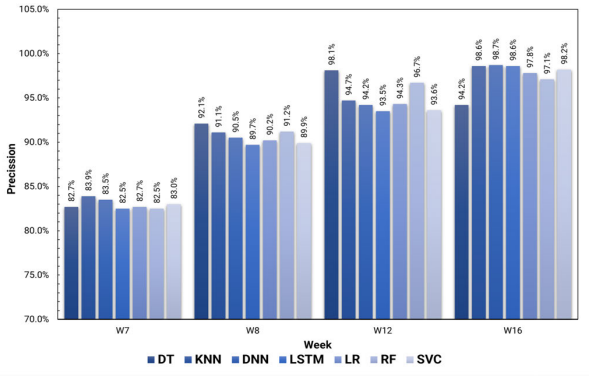


FIGURE 12. Precision of predictive models distributed by weeks with SMOTE.

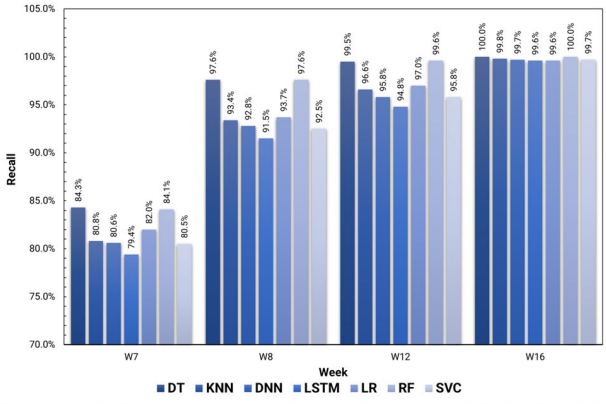


FIGURE 13. Recall of predictive models distributed by weeks with SMOTE.

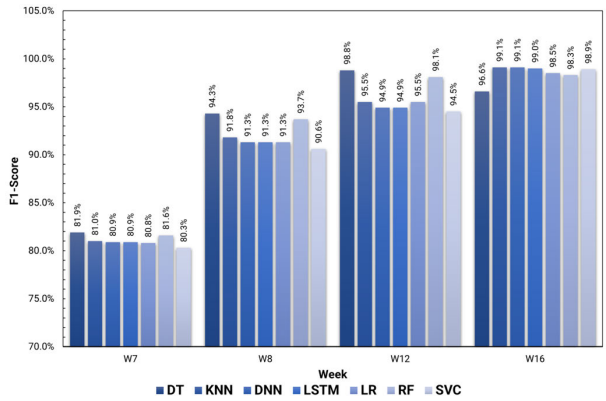


FIGURE 14. F1-Score of predictive models distributed by weeks with SMOTE.

However, in week 16, it drops to 96.6%. Likewise, in week 16, KNN, DNN, and LSTM with SMOTE are presented as the best classifiers with 99%. Fig. 14 displays the results of the F1-Score measure with SMOTE.

b: PERFORMANCE EVALUATION OF THE MODELS OF MACHINE LEARNING USING THE GAN TECHNIQUE

In Fig. 15 the results of the seven predictive models based on their accuracy using GAN to classify two categories (pass and fail) are presented. It can be observed that LSTM achieves

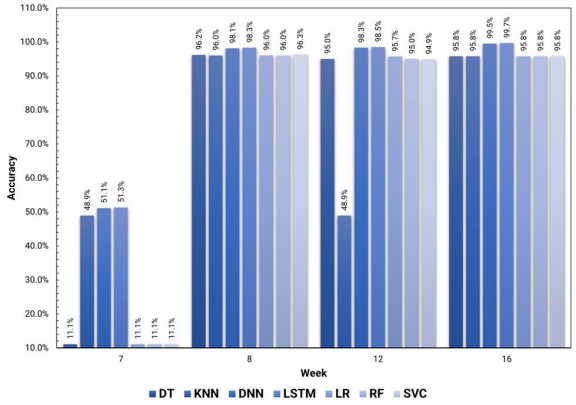


FIGURE 15. F1-Score of predictive models distributed by weeks with GAN.

accuracies of 51.3%, 98.3%, 98.5%, and 99.77% in weeks 7, 8, 12, and 16, respectively. Another notable predictive model is DNN, which achieves accuracies of 51.1%, 98.1%, 98.3%, and 99.5% in weeks 7, 8, 12, and 16.

Regarding the average precision, LSTM with GAN shows precision values of 65.8%, 98.3%, 98.5%, and 99.7% in weeks 7, 8, 12, and 16, indicating it learned to mitigate the error of false positives Fig. 16 displays the results of the precision measure for predictive models with GAN.

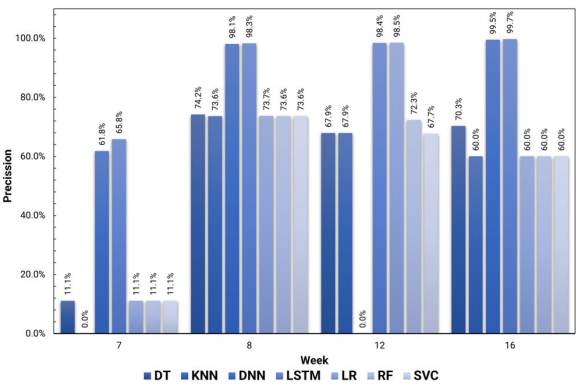


FIGURE 16. Precision of algorithmic models distributed by weeks with GAN.

The recall and F1-score measures follow the same classification trend, as shown in Figs. 17, and 18.

c: THE PERFORMANCE OF CLASSIFIERS SPECIFICALLY IN WEEK 8

Fig. 19 summarizes the evaluation of the predictive models in week 8 according to their accuracy. DT achieves 93.4% accuracy using the original data and 93.1% accuracy using SMOTE as a data balancing method. When applying GAN to balance the data, we can see that LSTM achieves an accuracy of 98.3%.

Fig. 20 summarizes the evaluation of the predictive models in week eight based on their precision. DT achieves 92.1% accuracy using the original data and 92.1% using SMOTE as

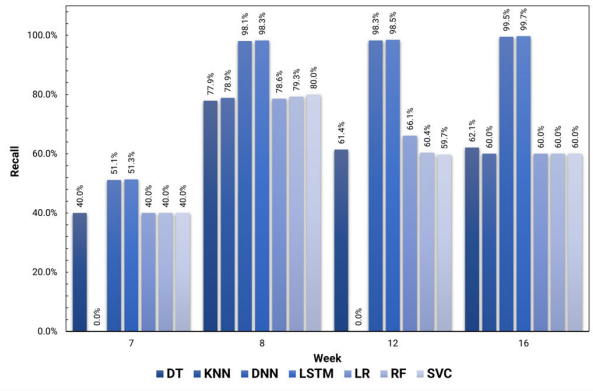


FIGURE 17. Recall of predictive models distributed by weeks with GAN.

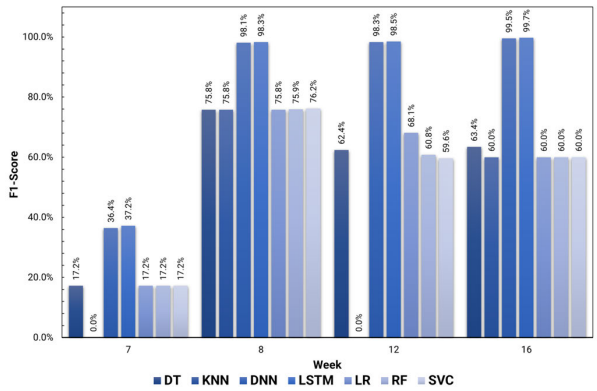


FIGURE 18. F1-Score of predictive models distributed by weeks with GAN.

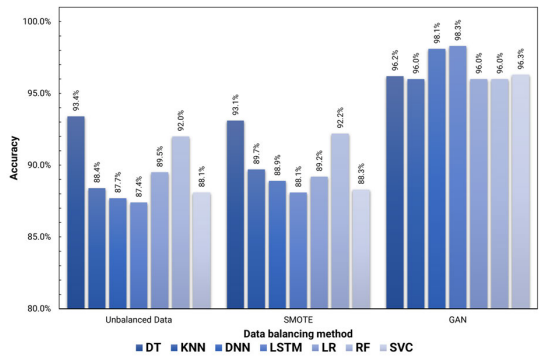


FIGURE 19. Accuracy obtained by each predictive model at week 8.

a data balancing method. However, it can be observed that traditional predictive models using GAN achieve precision measures lower than 75%, while DNN and LSTM reach 98.1% and 98.3% precision, respectively, when using GAN.

Fig. 21 presents the evaluation results of the recall measure for the seven predictive models in week 8. DT and RF achieve 97.9% and 97.8% using the original data, respectively. Similarly, DT and RF attain a recall of 97.6% using SMOTE. However, when using GAN, the traditional predictive models show an increased error percentage in

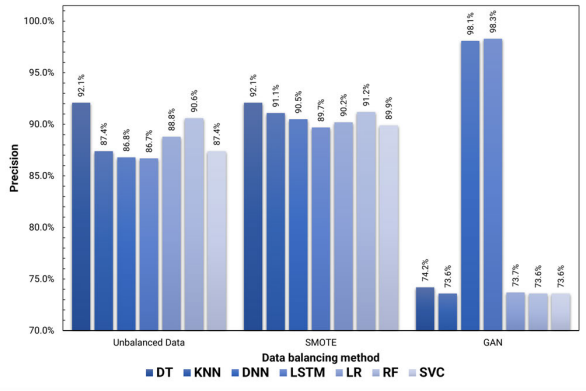


FIGURE 20. The precision obtained by each predictive model at week 8.

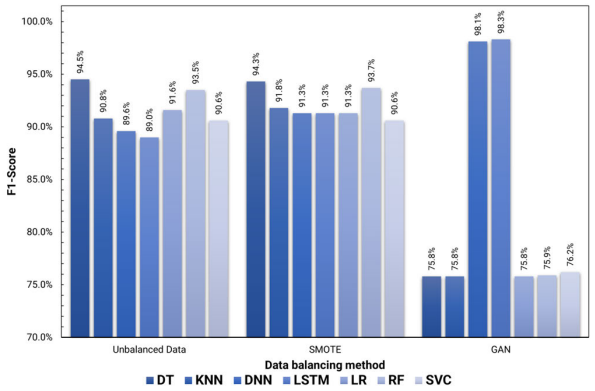


FIGURE 22. The F1-Score obtained by each predictive model at week 8.

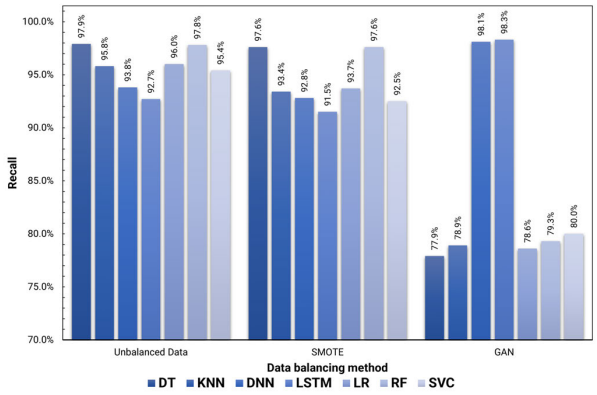


FIGURE 21. The recall obtained by each predictive model at week 8.

classifying false negatives, while DNN and LSTM, using GAN, achieve 98.1% and 98.3% recall.

Regarding the F1-Score measure, DT and RF show better percentages with original data, 94.5% and 93.5%, respectively. Using SMOTE, they achieved 94.3% and 93.7% in week 8, respectively. However, both measures drop to 75.8% and 75.9% using GAN, implying that the classifier error using GAN is 24.2% and 24.1%, respectively. In contrast, DNN and LSTM achieve 98.1% and 98.3% using GAN, respectively. Fig. 22 presents the results of the predictive models using data balancing methods in week 8.

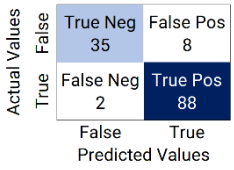
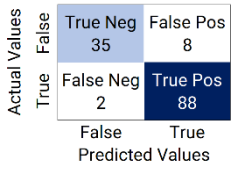
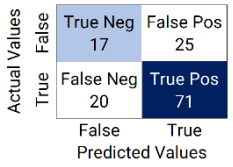
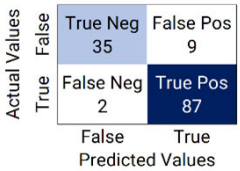
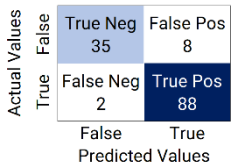
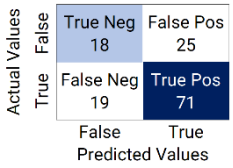
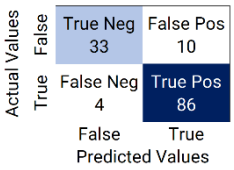
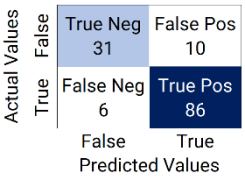
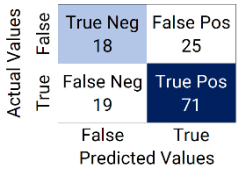
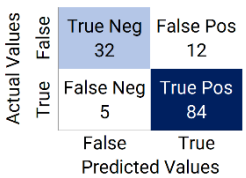
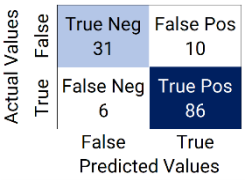
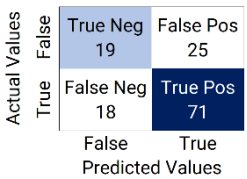
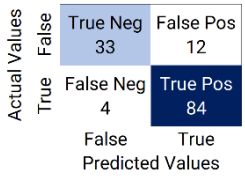
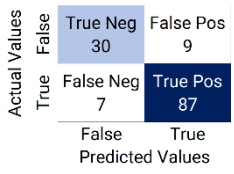
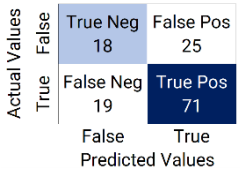
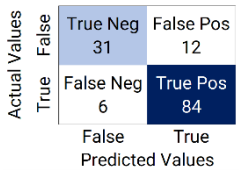
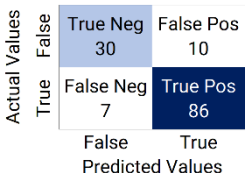
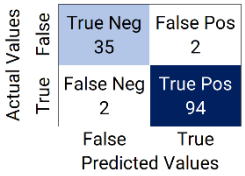
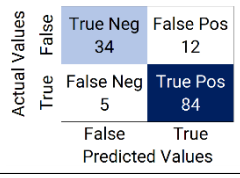
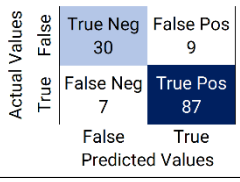
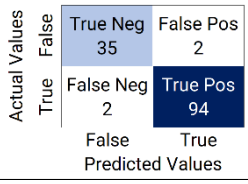
Table 7 presents the confusion matrix of each predictive model (DT, KNN, DNN, LSTM, LR, RF, SVC) with respect to the application of data balancing methods (unbalanced data, SMOTE, GAN) distributed in week 8. The confusion matrix allows for the derivation of accuracy, recall, precision, and F1-Score measures, as well as the classification error of each model. We can conclude that LSTM and DNN, both using GAN, manage to predict two students, according to their academic performance, as false positives and two students, according to their academic performance, as false negatives, resulting in a 3% error rate in classification. However, the predictive model DT achieves a 34% error rate in classification, meaning this model predicts 25 students, according to their academic performance, as false positives and 20 students as false negatives. According to the analysis, the data generated by SMOTE, and the original data present a similar distribution; however, the data generated by GAN show floating-point values with decimal parts ranging from 1 to 8 decimals. GAN presents data that adapts to deep learning predictive models such as LSTM and DNN, but traditional models do not train adequately with these data. Likewise, we can observe data distribution according to the attributes and the original data generated by SMOTE and GAN, as shown in Table 8.

Table 9 shows the values obtained by the sensitivity measure to evaluate the models' ability to predict a passing student as a true positive. With unbalanced data and SMOTE, LSTM shows a lower percentage of 93% and 92%, respectively. However, using GAN, LSTM presents a better ratio of true positives at 98% compared to other predictive models. Similarly, the specificity measure demonstrates the models' ability to predict a failing student as a true negative. LSTM shows the lowest percentages of 72% and 75% in unbalanced data and SMOTE, respectively; however, LSTM better classifies failing students with a percentage of 95% using GAN.

According to the Receiver Operating Characteristic (AUC) measure, all predictive models show values above 0.5, indicating that they have learned to predict or classify student performance as positive, with the outcome being approved. RF obtains the best value with 96% and the lowest by DT with 92%. On the other hand, we have the AUCs obtained by applying the SMOTE data balancing method, with KNN, LSTM, and DNN showing the best results. Finally, the evaluation of AUC applying GAN highlights LSTM as the predictive model that best classifies true positives with a percentage of 87%.

The training time in milliseconds used by predictive models employing data balancing methods is also presented. LSTM requires a longer training time with unbalanced data (63.45 ms). DNN shows a longer training time with SMOTE (60.18 ms) and GAN (44.22 ms); however, for GAN's training time, the data generation time for synthetic

TABLE 7. Confusion matrix of each predictive model according to the method of balancing data obtained in the evaluation of week 8.

Model	Data balancing method: Unbalanced data	SMOTE	GAN
DT			
RF			
LR			
SVC			
KNN			
LSTM			
DNN			

data (449 ms) must be added, as GAN uses a discriminator and a synthetic data generator that are trained within another neural network. We can conclude that GAN requires a longer training duration.

Table 10 presents the results of the classification error measure for week 8.

Table 11 presents the results obtained by applying the Bonferroni-Holm Correction [41] for pairwise comparisons between our proposed model (LSTM) and other predictive models. The null hypothesis of the non-parametric test is that the means of the algorithm results based on the F1-Score with the application of stratified 5-fold cross-validation are the same, with a significance level of 0.05. It is demonstrated that the null hypothesis was rejected when comparing LSTM with SVC, KNN, and LR, indicating significant differences between the means of F1-Score values among these models. However, the null hypothesis is accepted when comparing LSTM with DT, DNN, and RF, as these models have no significant differences between the F1-Score values.

TABLE 8. Data generation according to the data balancing method.

Data Balancing method	Attributes												
	Y	C	G	QP1	PT	ME	QP2	FT	PC	FE	LC	M	FA	T
Unbalanced Data	2	1	0	15	20	20	20	20	18	14	20	19	17.90	Pass
	2	1	1	6	0	0	5	6	14	6	11	10	5.05	Fail
SMOTE	1	2	0	6	15	6	8	16	16	9	16	15	10.50	Fail
	1	2	0	13	18	8	12	18	16	11	18	14	13.25	Pass
GAN	1	0	0	16.9287273	18.0377125	13.917263	10.192746	12.0377125	17.02833	6.07176373	17.9263617	18.9736284	11.59	Fail
	1	1	0	11.927283	19.1543692	13.5744	10.5906154	14.9387373	17.837383	8.59061538	18.642762	16.9872737	12.55	Pass

Y=Year of Income; C=Career; G=Gender; QP1=Qualified Practice 1; PT=Partial Task; ME=Midterm Exam; QP2=Qualified Practice 2; FT=Final Task; PC=Participation in Class; FE=Final Exam; LC=Linguistic Comprehension; M=Mathematic; T=Target

TABLE 9. Measurements of sensitivity, specificity, AUC, and training time of the predictive models according to the data balancing method, applied to week 8.

Model	(Sensitivity)			(Specificity)			AUC			Training Time		
	Unbalanced data	SMOTE	GAN	Unbalanced data	SMOTE	GAN	Unbalanced data	SMOTE	GAN	Unbalanced data	SMOTE	GAN
DT	0.98	0.98	0.78	0.81	0.81	0.40	0.929	0.935	0.85	0.012	0.03	0.02
RF	0.98	0.98	0.79	0.80	0.81	0.42	0.96	0.963	0.856	0.467	0.44	0.50
LR	0.96	0.93	0.79	0.77	0.76	0.42	0.95	0.95	0.83	0.975	0.06	1.21
SVC	0.94	0.93	0.80	0.73	0.76	0.43	0.94	0.95	0.83	0.063	2.52	1.11
KNN	0.95	0.93	0.79	0.73	0.77	0.42	0.95	0.96	0.83	0.091	0.11	0.08

TABLE 10. Classification error of the predictive models according to the data balancing method, applied to week 8.

Model	Classification Error		
	Unbalanced data	SMOTE	GAN
DT	8%	8%	34%
RF	8%	8%	33%
LR	11%	13%	33%
SVC	13%	13%	32%
KNN	12%	13%	33%
LSTM	14%	14%	3%
DNN	13%	13%	3%

TABLE 11. Bonferroni-Holm correction test results.

Comparison	5-fold cross-validation		
	Statistic	p-value	Result
LSTM vs SVC	2.48	0.00015	H0 is rejected
LSTM vs KNN	3.31	0.00034	H0 is rejected
LSTM vs LR	3.21	0.00042	H0 is rejected
LSTM vs DT	1.57	0.18870	H0 is accepted
LSTM vs DNN	1.25	0.13503	H0 is accepted
LSTM vs RF	1.021	0.12277	H0 is accepted

V. CONCLUSION AND FUTURE WORK

In this study, we present the results and findings from the articles that make up the research on predicting students' academic performance in the fundamentals of programming courses using LSTM and EDM.

In this research, we used 667 records from the fundamentals of programming course of the Computer Science degree and related fields from two Peruvian universities. After data cleaning, we achieved 661 records with thirteen attributes comprising our models' input.

We included six additional predictive models: Deep Neural Network, Decision Tree, Random Forest, Logistic Regression, Support Vector Machine, and K-Nearest Neighbor. We addressed the problem of unbalanced and small data using data balancing techniques, such as SMOTE and GAN.

The performance of the proposed models with unbalanced data was evaluated. The results show that traditional models such as Decision Tree and Random Forest achieve accuracy between 77% (week 7) and 99.9% (week 16), but with rates of false positives and false negatives due to the bias of the predominant class (approved). Likewise, it was found that the performance of the models using balanced data shows better values in terms of their precision and recall, with DNN and LSTM being the models with the highest precision and recall.

Week 8 is strategic for organizational decision-making regarding the Programming Fundamentals course. In that context, LSTM with GAN presents an accuracy, recall, precision, and F1-Score of 98.3%, followed by DNN-GAN with 98.1%.

The results show that SMOTE generates better results than GAN as a balancing method. This is because SMOTE adds synthetic data from vector space with fewer variations. However, GAN creates more realistic synthetic samples that are different from each other. The above facilitates the models to learn the SMOTE data and overfit. The experiments with GAN generalize with the models.

GAN has proven to adapt to the requirements and objectives of this research. However, the analysis of the data generated by GAN was done manually, finding 64 inconsistent data, which were eliminated.

For future research, more data should be considered, and attributes related to competencies and learning styles should be incorporated. Likewise, it is expected to be

able to predict a student's dropout using Graph Neural Networks (GNN) and incorporate other data balancing techniques, like undersampling or hybrid methods.

ACKNOWLEDGMENT

To the Research Directorate of the Peruvian University of Applied Sciences for the support provided to realize this research work.

REFERENCES

[1] F. A. Delgado and A. I. Martel, "II informe bienal sobre la realidad universitaria en el Perú," Perú, 2020. [Online]. Available: https://cdn.www.gob.pe/uploads/document/file/1230044/InformeBienal.pdf?v=1603336820

[2] "Computing curricula 2020: Paradigms for global computing education," ACM IEEE, 2020, doi: 10.1145/3467967.

[3] H. Nieto-Chaupis and A. Alfaro-Acuña, "The management of a private Peruvian university at pandemic times: Assessment of decisions and implications on the key indicators," in *Proc. 14th Int. Conf. Educ. Technol. Comput.*, Oct. 2022, pp. 555–560.

[4] O. Chamorro-Atalaya, S. Trujillo-Pérez, E. Perez-Linares, A. Torres-Quiroz, Y. Medina-Bedón, L. Quevedo-Sánchez, M. Fierro-Bravo, and A. Leva-Apaza, "Failure rate in university students: Analysis of its variation in the transition from face-to-face education to virtual education," *Int. J. Inf. Educ. Technol.*, vol. 13, no. 1, pp. 151–157, 2023.

[5] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming—12 years later," *ACM Inroads*, vol. 10, no. 2, pp. 30–35, 2019.

[6] C. Wang, J. Dai, and L. Xu, "Big data and data mining in education: A bibliometrics study from 2010 to 2022," in *Proc. 7th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2022, pp. 507–512.

[7] D. A. Shafiq, M. Marjani, R. A. A. Habeeb, and D. Asirvatham, "Student retention using educational data mining and predictive analytics: A systematic literature review," *IEEE Access*, vol. 10, pp. 72480–72503, 2022.

[8] Rimpy, A. Dhankhar, and K. Solanki, "Educational data mining tools and techniques used for prediction of student's performance: A study," in *Proc. 10th Int. Conf. Rel., Infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, Oct. 2022, pp. 1–5.

[9] Y. A. Alsariera, Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, and N. Ali, "Assessment and evaluation of different machine learning algorithms for predicting student performance," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, May 2022.

[10] Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, Y. A. Alsariera, A. Q. Ali, W. Hashim, and S. K. Tiong, "Toward predicting student's academic performance using artificial neural networks (ANNs)," *Appl. Sci.*, vol. 12, no. 3, p. 1289, Jan. 2022.

[11] R. Ordoñez-Avila, N. Salgado Reyes, J. Meza, and S. Ventura, "Data mining techniques for predicting teacher evaluation in higher education: A systematic literature review," *Heliyon*, vol. 9, no. 3, Mar. 2023, Art. no. e13939.

[12] S. Chinchua, T. Kantathanawat, and S. Tuntiwongwanich, "Increasing programming self-efficacy (PSE) through a problem-based gamification digital learning ecosystem (DLE) model," *J. Higher Educ. Theory Pract.*, vol. 22, no. 9, pp. 131–143, 2022.

[13] A. Sharma, P. K. Singh, and R. Chandra, "SMOTified-GAN for class imbalanced pattern classification problems," *IEEE Access*, vol. 10, pp. 30655–30665, 2022.

[14] M. Scott and J. Plested, "GAN-SMOTE: A generative adversarial network approach to synthetic minority oversampling for one-hot encoded data," in *Proc. ICONIP*, 2019, vol. 15, no. 2, pp. 29–35.

[15] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Comput. Electron. Agricult.*, vol. 200, Sep. 2022, Art. no. 107208.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[17] J. Malini and Y. Kalpana, "Analysis of factors affecting student performance evaluation using education dataminig technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 7, pp. 2413–2424, 2021.

[18] C. Cabo, "Use of machine learning to identify predictors of student performance in writing viable computer programs with repetition loops and methods," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2021, pp. 1–9.

[19] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, "Transfer learning from deep neural networks for predicting student performance," *Appl. Sci.*, vol. 10, no. 6, p. 2145, Mar. 2020.

[20] I. Sandoval-Palis, D. Naranjo, R. Gilar-Corbi, and T. Pozo-Rico, "Neural network model for predicting student failure in the academic leveling course of Escuela Politécnica Nacional," *Frontiers Psychol.*, vol. 11, pp. 1–7, Dec. 2020.

[21] A. Farissi, H. Mohamed Dahlan, and Samsuryadi, "Genetic algorithm based feature selection with ensemble methods for student academic performance prediction," *J. Phys., Conf. Ser.*, vol. 1500, no. 1, Apr. 2020, Art. no. 012110.

[22] M. Akour, H. A. Sghaier, and O. Al Qasem, "The effectiveness of using deep learning algorithms in predicting students achievements," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 388–394, Jul. 2020.

[23] N. L. Alvarez, Z. Callejas, and D. Griol, "Predicting computer engineering students' dropout in cuban higher education with pre-enrollment," *J. Technol. Sci. Educ.*, vol. 10, no. 2, pp. 241–258, 2020.

[24] F. A. Al-Azazi and M. Ghurab, "ANN-LSTM: A deep learning model for early student performance prediction in MOOC," *Heliyon*, vol. 9, no. 4, Apr. 2023, Art. no. e15382.

[25] A. Kukkar, R. Mohana, A. Sharma, and A. Nayyar, "Prediction of student academic performance based on their emotional wellbeing and interaction on various e-learning platforms," *Educ. Inf. Technol.*, vol. 28, no. 8, pp. 9655–9684, Aug. 2023.

[26] M. Yağcı, "Educational data mining: Prediction of students' academic performance using machine learning algorithms," *Smart Learn. Environ.*, vol. 9, no. 1, pp. 9–11, Dec. 2022.

[27] D. Alboaneen, M. Almelihi, R. Alsubaie, R. Alghamdi, L. Alshehri, and R. Alharthi, "Development of a Web-based prediction system for students' academic performance," *Data*, vol. 7, no. 2, p. 21, Jan. 2022.

[28] Md. M. Rahman, Y. Watanobe, T. Matsumoto, R. U. Kiran, and K. Nakamura, "Educational data mining to support programming learning using problem-solving data," *IEEE Access*, vol. 10, pp. 26186–26202, 2022.

[29] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Prediction of students' academic performance based on Courses' grades using deep neural networks," *IEEE Access*, vol. 9, pp. 140731–140746, 2021.

[30] N. R. Aljohani, A. Fayoumi, and S.-U. Hassan, "Predicting at-risk students using clickstream data in the virtual learning environment," *Sustainability*, vol. 11, no. 24, pp. 1–12, Dec. 2019.

[31] N. Wu, L. Zhanh, Y. Gao, M. Xhang, X. Sun, and J. Feng, "CLMS-Net : Dropout prediction in MOOCs with deep learning," in *Proc. ACM TURC*, no. 75, 2019, p. 6.

[32] S. Hassan, H. Waheed, N. R. Aljohani, M. Ali, S. Ventura, and F. Herrera, "Virtual learning environment to predict withdrawal by leveraging deep learning," *Int. J. Intell. Syst.*, vol. 34, no. 8, pp. 1935–1952, Aug. 2019.

[33] S. Altaf, W. Soomro, and M. I. M. Rawi, "Student performance prediction using multi-layers artificial neural networks: A case study on educational data mining," in *Proc. 3rd Int. Conf. Inf. Syst. Data Mining*, Apr. 2019, pp. 59–64.

[34] A. Almasri, E. Celebi, and R. S. Alkhawaldeh, "EMT: Ensemble meta-based tree model for predicting student performance," *Sci. Program.*, vol. 2019, pp. 1–13, Feb. 2019.

[35] V. A. Nguyen, Q. B. Nguyen, and V. T. Nguyen, "A model to forecast learning outcomes for students in blended learning courses based on learning analytics," in *Proc. 2nd Int. Conf. E-Soc., E-Educ. E-Technol.*, Aug. 2018, pp. 35–41.

[36] J. Park, S. Kwon, and S.-P. Jeong, "A study on improving turnover intention forecasting by solving imbalanced data problems: Focusing on SMOTE and generative adversarial networks," *J. Big Data*, vol. 10, no. 1, pp. 1–16, Mar. 2023.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[38] S. H. Rafi, Nahid-Al-Masood, S. R. Deeba, and E. Hossain, "A short-term load forecasting method using integrated CNN and LSTM network," *IEEE Access*, vol. 9, pp. 32436–32448, 2021.

[39] X. Chen, W. Chen, V. Dinavahi, Y. Liu, and J. Feng, "Short-term load forecasting and associated weather variables prediction using ResNet-LSTM based deep learning," *IEEE Access*, vol. 11, pp. 5393–5405, 2023.

[40] H. H. Goh, B. He, H. Liu, D. Zhang, W. Dai, T. A. Kurniawan, and K. C. Goh, "Multi-convolution feature extraction and recurrent neural network dependent model for short-term load forecasting," *IEEE Access*, vol. 9, pp. 118528–118540, 2021.
[41] H. Abdi, "Holm's sequential Bonferroni procedure," *Encyclopedia Res. Des.*, vol. 1, no. 8, pp. 1–8, 2010.

LUIS VIVES was born in Chiclayo, Peru, in 1984. He received the bachelor's degree in systems engineering from Universidad Señor de Sipan, in 2006. He is currently pursuing the Ph.D. degree in engineering with Pontificia Universidad Catlóica del Perú. He is also a Professor of computer science with Universidad Peruana de Ciencias Aplicadas. His research interests include software engineering and computer science, software quality, software architecture, machine learning, and knowledge extraction based on the semantic web.

IVAN CABEZAS (Member, IEEE) was born in Ibagué, Colombia, in 1973. He received the bachelor's degree in systems engineering and the Ph.D. degree in computer sciences from Universidad del Valle, Cali, Colombia, in 2004 and 2013, respectively. He is currently a Professor of computing and smart systems with Universidad ICESI. His research interests include software architecture, cybersecurity, blockchain, pattern recognition, and outcome-based engineering education.

JUAN CARLOS VIVES was born in Chiclayo, Peru, in 1982. He received the bachelor's degree in mechanical engineering from César Vallejo University, Trujillo, Perú, in 2009, and the master's degree in mechanical engineering from the National University of Trujillo-Peru, in 2017, where he is currently pursuing the Ph.D. degree in science and engineering. He is currently the Director of the Professional School of Electrical Mechanical Engineering at the Señor de Sipan University. His research interests are the design of electromechanical and power equipment using composite materials and computer aided design (CAD) and finite element analysis (FEA) software.

NILTON GERMAN REYES was born in Otuzco-Peru. He received the graduate degree in computer and systems engineer from the Antenor Orrego Private University, Trujillo, the master's degree in business administration with a Mention in Business Management from the Pedro Ruiz Gallo National University of Lambayeque, and the Ph.D. degree in education from César Vallejo University. He is currently work as a Full-Time Principal Professor at the Pedro Ruiz Gallo National University of Lambayeque. His areas of interest are software engineering and computer science. His research interests are software quality, machine learning, digital transformation, and artificial intelligence.

JANET AQUINO was born in Piura, Peru. She received the degree in computer and systems engineer from Antenor Orrego Private University, Trujillo, the master's degree in business administration with a Mention in Business Management from the Pedro Ruiz Gallo National University of Lambayeque, and the Ph.D. degree in computer science from the Señor de Sipan University. She is currently pursuing the Ph.D. degree in education with César Vallejo University. She is currently works as a full-time Associate Professor at the Pedro Ruiz Gallo National University of Lambayeque and a part-time Professor at the Technological University of Peru.

JOSE BAUTISTA CÓNDOR received the master's degree in education with a mention in Educational Psychology from the National University of Trujillo-Peru, 1995. He is a Doctor of Child Psychology at César Vallejo University, Trujillo, Peru, 2013. He is currently works as a Teacher at the National University of Trujillo. His research interest includes develop educational programs to improve learning.

S. FRANCISCO SEGURA ALTAMIRANO received the Bachelor of Electronic Engineering degree from Universidad Nacional Mayor de San Marcos, in 2001. He was with various telecommunications companies in the area of fiber optic networks, until 2010. He is currently a Teacher of electronic engineering specialty with Pedro Ruiz Gallo National University.