SuccessNet: An Automated Approach to Predict Student Academic Performance Using PCA Extracted CNN Novel Features and RF-SVM Ensemble Model

Khaled Alnowaiser[1] and Muhammad Umer[2,*]

[1]Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, P.O. Box 151, 11942, Saudi Arabia
[2]Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan
Email: k.alnowaiser@psau.edu.sa (K.A.); umer.sabir@iub.edu.pk (M.U.)
[*]Corresponding author

Manuscript received March 15, 2025; revised March 31, 2025; accepted April 29, 2025; published November 10, 2025

***Abstract*—With the rapid growth of data in the education sector, traditional techniques have failed to predict student academic success effectively. This research work uses features extracted from Convolutional Neural Networks (CNN) with a Random Forest (RF) and Support Vector Machine (SVM) ensemble model to predict the academic performance of students. We called this novel framework SuccessNet. It obviates manual feature extraction and surpasses independent Deep Learning (DL) and Machine Learning (ML) models in performance. The experiments are carried out in two sets. First, the original features are used to apply nine ML algorithms. The second set of experiments contains features extracted by CNN. The SuccessNet is formed with a soft voting mechanism that combines the top models generated during the above two sets of experiments based on academic performance prediction for students using an ensemble of RF and SVM. A comparison of performance with existing models shows auspicious results. SuccessNet gives an accuracy of 99.35% with a precision, recall, and F-Score of 99%.**

***Keywords*—computer and education, educational data mining, ensemble learning, machine learning**

I. Introduction

The COVID-19 pandemic has profoundly disrupted traditional educational systems, significantly reducing in-person classes and necessitating a widespread shift to online learning environments. In this regard, UNESCO requested that education institutions invest in digital resources that could support an appropriate transition. However, online schooling is gaining momentum and yet has several issues, mainly related to content deliverance and technological support [1]. Almaiah *et al*. [2] view acceptance of students as a crucial component of online learning systems success. Al-Arabi *et al*. point out the importance of technological factors for system effectiveness [3].

These challenges have prompted higher education institutions to adopt digital platforms such as Learning Management Systems (LMS) and Massive Open Online Courses (MOOCs) to facilitate instruction and manage educational content. However, as institutions increasingly rely on these digital tools, concerns have emerged about their ability to provide comprehensive insights into student learning behaviors and academic progress. Educators have frequently pointed out that data generated from LMS usage is often insufficient to reflect true learning outcomes or to detect performance anomalies effectively. As a result, students resort to using multiple platforms to supplement their learning experience, making it difficult to assess academic performance through LMS data alone.

Given this landscape, there is a growing need for intelligent, automated systems capable of predicting student academic success by analyzing a diverse range of data sources. This study addresses this gap by proposing an advanced machine learning-based framework SuccessNet that leverages convolutional neural networks and ensemble learning techniques to provide accurate academic performance predictions. By doing so, the proposed approach not only helps institutions proactively identify at-risk students but also contributes to improving online learning outcomes in an era increasingly defined by digital education.

Data mining is extracting meaningful patterns and insights from extensive data. It is primarily used in most domains, including manufacturing [4], healthcare, finance, and education [4–6]. As technology grows, more data is generated in education that needs to be analyzed to make better decisions [7]. In particular, Educational Data Mining (EDM) is an area that deals with raw data stemming from educational information in an attempt to predict student performance and discover actionable improvements. Applying such raw educational data to EDM models will lead to inferences that result in correct predictions and informed decisions.

This is where researchers specifically seek student-related information that has the potential to help predict dropouts, academic performance, off-task behavior, and even real-time monitoring of psychological well-being through sensors and Wearable [8, 9]. Most of the literature available at this point about predicting students' academic performance relies on the term EDM. However, they predominantly rely on demographic information and online activities. How students interact with videos in video-assisted learning environments has been sporadically analyzed by a few research studies [10]. The current study is thus an attempt to estimate the way students perform in virtual video classrooms through analysis of data from platforms like eDify, Moodle, and Student Information System (SIS), which will help optimize the teaching and learning process.

Predicting students' academic performance through Machine Learning (ML) brings about early detection of a student at risk [11]. In this way, timely intervention and personalized support can be given. This will help the educator quickly change their teaching strategies to make them more appropriate for better learning. Again, it supports proper resource allocation in place, guiding support services toward areas of most need [12]. These ML models enable an in-depth analysis of student data patterns and, therefore, provide actionable insights to foster an environment that gives every student an equal chance at success [13, 14]. Ultimately, educational quality is awarded, with better, more equitable opportunities for all students to have quality learning. Following are the major contributions of the proposed framework for student academic success prediction.

- In this paper, we proposed a framework called SuccessNet that incorporates features extracted from a modified Convolutional Neural Networks (CNN) with Random Forest (RF) and Support Vector Machine (SVM) ensemble models in predicting students' academic performance. The Principal Component Analysis (PCA) is utilized to extract 326 significant features from CNN-generated features.
- This research work also explains how complex features affect the ML models. For this, models are trained and tested on original and convoluted features and checked for performance comparison.
- ML algorithms like SVM, K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), Decision Tree (DT), Extra Tree Classifier (ETC), RF, Gradient Boosting Machine (GBM), and Logistic Regression (LR) are used to compare performances.
- Finally, this study evaluates the proposed model against recent methods and compares them in metrics based on recall, accuracy, F1-Score, and precision.

The remainder of the study is organized as follows: Section II presents the literature review of recent works predicting the academic achievement of students through ML. Section III outlines the dataset and methodology proposed, detailing classifiers in ML to be employed and evaluation parameters. Section IV reports the experimental results and relevant discussions. Finally, Section V concludes the study with suggestions for future research.

II. Related Works

In several domains, ML and Deep Learning (DL) have lately gained popularity as categorization and prediction methods. EDM has also gained popularity in education to forecast or predict student performance, and researchers are seeking ways to improve student performance. This section will provide a glimpse into the existing research conducted, finding the influence on student academic EDM achievements. Many research investigations have been undertaken on the aspect of student performance anticipation, considering various perspectives and influencing factors. Sarwat *et al*. [15] applied a Conditional Generative Adversarial Network (CGAN) in coordination with a deep SVM to forecast the academic achievement of students. The researchers created artificial data samples using an enhanced CGAN, which they implemented to study the performance using a relatively small dataset. The validation of the proposed system was carried out with experiments under scenarios with and without CGAN. Different kernel-based techniques like sigmoid, polynomial, radial, and linear functions were used to test the deep SVM. From the research, the results obtained demonstrate how the deep SVM model in the CGAN is better than any other model concerning specificity, sensitivity, and area under the curve.

Kaunang and Rotikan [16] developed a prediction model, an ML-based system for the prediction of a student's performance in studies. The prediction framework was tested with a dataset from a questionnaire tapping various demographic details, Grage Point Average (GPA) from the previous semester, and family history. The authors tested the model with RF and DT algorithms and could get an accuracy of up to 69.9%. Hernandez *et al*. [17] designed A method utilizing DL to predict students' academic achievement in higher education institutions in both public and private domains, irrespective of gender, in Colombia. Recent research was done to implement a systematic procedure for using Artificial Neural Networks in analyzing the significance level of various predictors of academic performance in universities. The study below considered Artificial Neural Network (ANN) for predicting students' performance with around 82% accuracy.

Similarly, in another study, a DL model was developed for predicting the academic achievement of pupils using information from two courses [18]. The dataset used for this research was most imbalanced, and Synthetic Minority Oversampling Technique (SMOTE) was employed with many other techniques in this respect. The ML models' performance was compared over multiple feature sets. The experimental outcomes illustrated how good the performance of DL models was against those data For the Portuguese course dataset, there was 96.4% accuracy and 99% precision for the mathematics course dataset, there was 93.2% accuracy and 94% precision. Apart from this, Alberto *et al*. [19] applied the RF, eXtreme Gradient Boosting (XGB), DT, and Multi-Layer Perceptron (MLP) classifiers for student performance prediction. Based on this, the accuracy score reached 78.2%, much better than classifiers in that domain.

In recent studies, various ML techniques have been employed to predict student academic performance and address related issues. Kuadey *et al*. [11] examined the impact of technostress on student learning burnout using ML algorithms. Hussain *et al*. [12] utilized data mining approaches to predict academic performance, demonstrating the effectiveness of data-driven methods. Umamaheswari *et al*. [13] introduced a modified SVM approach for student success prediction, highlighting improvements in predictive accuracy. Narayanan and Kumaravel [14] proposed a Chaotic Optimized Boost Long Short-Term Memory (LSTM) model, focusing on enhancing performance prediction in educational sectors through advanced neural network architectures. These studies collectively underscore the potential of ML in educational data analysis.

While existing research has demonstrated the potential of various machine learning and deep learning models in predicting student performance, many lack robust feature extraction strategies or ignore the effects of high-dimensional data on prediction accuracy. Additionally, few studies have evaluated ensemble models combining traditional ML and deep learning features, especially under real-world data imbalance. Our proposed framework, SuccessNet, fills this gap by integrating CNN-based feature extraction with PCA for dimensionality reduction and employing a soft voting ensemble of RF and SVM. This approach not only boosts classification accuracy but also improves generalizability and interpretability, as demonstrated by superior results across multiple performance metrics. The limitations of previous research work are summarized in Table 1.

Table 1. Summary of related work and identified limitations

Ref.	Methods	Dataset	Results	Limitations
[15]	CNN, Deep SVM, CGAN, LSTM, RF, LR, ETC, GBM, SGD	Cortez dataset	AUC = 92.36%	Used small synthetic dataset, lacks generalization across multiple institutions
[16]	RF, Decision Tree	Real-world data (Universitas Klabat)	DT accuracy = 66.9%	Low accuracy, limited feature engineering and ensemble testing
[17]	DT, ANN, GBT, RF, FLM, LR	ICFES data	ANN accuracy = 82%	Does not test CNN-based feature extraction or ensemble optimization
[18]	DL model	Cortez dataset	DL accuracy = 96.4%	Limited class balancing, lacks integration with dimensionality reduction
[19]	DT, NB, Rule-based	Universiti Sultan Zainal Abidin	Rule-based accuracy = 71.3%	Does not explore deep learning or PCA-based fusion
[9]	MLP, LDA, NB, SVM, KNN	British University Dubai	LDA accuracy = 79%	Small dataset, lacks CNN or hybrid models
[20]	CNN, ANN, KNN, DT, LR, NB	OULAD dataset	CNN accuracy = 88%	Focus on CNN alone, no feature re-duction applied
[21]	RF, NN, LR, SVM, NB, KNN	Turkish university	Accuracy between 70–75%	No hybrid or ensemble comparison shown
[22]	Firefly algorithm, ANN, MLPNN	SEU (Saudi Arabia)	RMSE = 0.39	Focuses only on error minimization, lacks classification strength
[23]	RF, NN, DT, XGB, SVM, SRM	IoT course (China)	SRM accuracy = 70.8%	Weak accuracy, no ensemble or CNN integration
[24]	SRL model	Business Communication (82 students)	Accuracy above 88%	Small sample size, lacks diverse dataset testing
[25]	DT, RF, SVM, LR, NN	Pennsylvania school	LR accuracy = 60%	Basic ML use only, lacks deep learning or PCA use
[26]	FCM, MLP, LR, RF, FCM-MLP, FCM-RF	Kaggle dataset	Accuracy = 95.83%	Fusion methods used but not explained in depth
[27]	LR, KNN, DT, RF, NB, SVM, DL	Self-collected data	RF accuracy = 85%	No CNN or feature reduction integration used

In another study on predicting student academic performance, Lubna *et al*. [9] explored the use of Naive Bayes, Latent Dirichlet Allocation (LDA), SVM, KNN, and Multilayer Perceptron ML, reporting that SVM gave the highest accuracy score of 76.3%. Poudyal *et al.* [20] that suggested a hybrid DL model based on academic success prediction for students suggested combining two 2D CNN models with the integration of ML techniques, including DT, NB, LR, and KNN. With an accuracy of 88%, the results showed that the suggested 2D CNN system was better than alternative ML models.

Chen and Ding [25] proposed an ML-based system for the prediction of the student's academic performance in Pennsylvania's Schools. In this study, the authors used ML and DL systems such as DT, RF, LR, SVMs, and NN. Results show that the Deep Neural Network achieved the highest performance of 60%. Mirza *et al.* [26] explored the concept of predicting student performance in education and its various iterations. They discussed several ML approaches, including Fuzzy C-Means (FCM), MLP, LR, and RF algorithms, for predicting student achievement in the classroom. The study thoroughly investigates both the currently used models for predicting student performance and the newly proposed ones in the paper. It examines different combinations of these algorithms, such as FCM-LR, FCM-MLP, and FCM-RF. The results indicate that FCM-MLP and FCM-LR achieved an accuracy of 95.833%. Badal and Sungkurhe [27] developed a framework to predict student grades in an online grading system. The authors utilized ML based techniques to predict the students' academic grades. Their best-performing model is RF, which gives an accuracy of 85%.

Furthermore, Yagci [21] utilized prediction frameworks based on trees and regression to forecast the performance of the 1,854 enrolled students in the Turkish Language I course, including NN, NB, RF, KNN, SVM, and LR. When considering three parameters using departmental, professor, and midterm exam results, for example, the study increased categorization accuracy from 70% to 75%. Identified the effect factors for predicting students' performance, Hamadneh *et al*. [22] added more detail by proposing a DL system to predict the students' academic achievement in an e-learning environment. They used one ANN for performance prediction in blended learning and carried out a statistical analysis to determine the variables influencing the students' performance. The Firefly-trained ANN model scored an RMSE of 0.39. Liu *et al*. [23] reported the design of a feedforward spike neural network introduced in predicting students' achievement at the college level, and the study received a 70.8% accuracy rating. On the other hand, Ali and Hanna [24] employed clustering algorithms to estimate the success of pupils enrolled in a course. The results showed that using engagement activities and log data to predict academic success was more accurate than 88% of the time. A summary of the research above is shown in Table 1. Thus, while each study described above has made an essential contribution to predicting student performance, leading to higher predictive accuracy, there is still room for improving the prediction accuracy of the models of academic performance.

III. Materials and Methods

This paper describes the dataset for identifying student academic success. Further, it explains the strategy and specifics of appropriate steps. It also includes a brief description of the classifiers used of ML.

A. Dataset

To predict how well pupils will achieve in their coursework, we have utilized the dataset obtained from the works of Hasan *et al.* [28], who compiled various types of structures of the dataset is collected using the MOODLE online platform. The final dataset is divided into three categories: student activity, academic data, and student video interaction. MOODLE was the source of student activity data. The comprehensive mapping of all the above results in a dataset with 21 features and a total of 326 instances. The options include "Prohibition", "CGPA", "Likes", "High_Risk", "Other_Modules", "ESE", "Plagiarism_history", "OnlineO", "CW1", "CW2", "Remote_Student", "Applicant_Name", "At Risk", "At_Risk_SSC", "OnlineC", "Paused", "Played", "Attempt_Count", "Term_Exceeded", "Segment", along with "Result". Table 2 represents a brief overview of the dataset, while Table 3 describes the dataset with details of each feature.

Table 2. Dataset statistics

Dataset	Information
Date Link	https://zenodo.org/records/5591907
Dataset Records	326
Data Features	21
Target Classes	Pass and Fail

B. Machine Learning (ML) Models

We develop machine-learning models using NLTK and Scikit-learn libraries. The supervised ML algorithms that we use in this study: DT, RF, KNN, LR, GNB, ETC, Stochastic Gradient Descent (SGD), GBM, and SVM. All these models are implemented using Python.

1) Random Forest (RF)

In the literature, RF is a vital tree-based ensemble model and is well adapted for usage in issues with classification [29]. By merging multiple weak learners, RF is able to attain the goal of very accurate predictions. The decision trees use a bagging process in the training phase. A bootstrap dataset of the same size as the original training dataset is produced through subsampling. When constructing decision trees, RF needs to choose the attributes for each level's root node. It is a part of the prediction process. This method is mainly known as attribute selection.

Table 3. Dataset description

Sr.	Attribute	Description
1	Applicant_Name	Includes the name and other personal student details
2	CGPA	Student's CGPA information.
3	Attempt_Count	Information about the number of attempts a student has taken for a specific subject/module.
4	RemoteStudent	Indicates whether the student attends classes remotely.
5	Probation	Backlog of modules that the student needs to clear.
6	HighRisk	Indicates the risk of failing a subject.
7	TermExceeded	Shows the rate of degree development.
8	AtRisk	Indicates if a pupil is at danger because they have failed a course in the past.
9	AtRiskSSC	Records any deficiencies registered by the Student Success Center (SSC).
10	Other Modules	Displays the student's current semester registration for further modules.
11	PlagiarismHistory	Keeps track of any plagiarism that occurs in any module.
12	CW1	Displays the student's grades from the first course assignment.
13	CW2	Displays the student's grades for the second course assignment.
14	ESE	Grades received on the final exam of the semester.
15	Online C	Embodies the student's activities while on campus.
16	Online_O	Extracurricular activities of student.
17	Played	Count of video played
18	Paused	Count of video paused
19	Likes	Count how many times video likes
20	Segment	Details about particular video portions that have been seen with the slider.
21	Result	The final result of the student (Target class).

2) Decision tree

The DT algorithm is considered a widespread and well-known method in ML to address issues with classification and regression [30]. It is commonly used as a rule-based and straightforward method of dividing a dataset using a binary method to divide it into smaller sets, progressively cutting the data until it cannot split, forming a kind of tree structure with branches of different sizes. It is followed by implementing maximum depth hyperparameters to prevent overfitting and simplify the model. A couple of the most common ways to make attribute selection for decision trees are using the "Gini Index" and "Information Gain".

3) K-Nearest Neighbor (KNN)

KNN [31] is a well-established ML technique that has been applied to problem-solving tasks with data classification. The KNN model discovers the K-values that indicate objects close to a point. Characteristic, the K-value aids in locating the closest feature vectors. The method tests for different quantities of K to come to ideal results. The nearest neighbors in the feature vector are calculated by KNN using the various distance metrics, which include Euclidean and Minkowski distances.

4) Logistic Regression (LR)

LR is a data analysis linear model that uses one or more variables to provide results [32, 33]. It serves as a regression technique with a statistical foundation that calculates the likelihood of class membership. In cases when the target variable is categorical, LR is especially recommended. LR uses the logistic function to estimate probabilities to clarify how categorical dependent factors and independent variables relate to one another. The logistic function, represented by a sigmoid curve, is calculated as:

$$f(x) = \frac{L}{1 + e^{-m(v - v_o)}}$$

5) Support Vector Machine (SVM)

SVM is a potent binary classification model task that provides flexibility using different kernel functions, as described in [15]. It classifies data points by finding a hyperplane within the feature space in which multiple features have different dimensionality this hyperplane. There are so many possible hyperplanes to choose from in an Ndimensional space that allow the best separation of data points into the two classes. The one-versus-all approach of SVM splits the data, continuously updating this process to divide the dataset into distinct classes. For data that cannot be separated, nonlinear SVMs are applied to change the coordinate space that was initially separable coordinate space. It can be directly formulated as $x = \phi(x)$.

6) Gradient Boosting Machine (GBM)

GBM [34] is a categorization task-specific ensemble model. It combines many weak learners, usually decision trees, to make predictions as accurately as possible. The weak learners are repeatedly trained on improved versions of the prior trees using the boosting process, which gradually improves their performance. Optimization of the coefficients of this implemented model is done by utilizing the loss function's gradient, measuring the correspondence of this model to the data set. Various parameters need to be tuned for optimal results with GBM to be achieved.

7) Extra Tree Classifier (ETC)

Another tree-based classifier in this study is the ETC. According to the approach in [35, 36], the Extra Trees Classifier simply takes the same principle as the decision tree and random forest with a slight modification of cutting the randomized tree at maximum output point. It creates trees on the whole sample, but at every tree node, it selects a random cut point instead of a piece-wise constant like in the RF. In this way, ETC can keep a multi-linear approximation. This is where ETC is superior to RF since the errors the learners make of the base learners are less correlated. Many studies point out that ETC outperforms RF in most cases. It extracts the features while building the tree, and it does data division by looking at the Gini index of each feature.

8) Gaussian Naive Bayes (GNB)

GNB belongs to a family of classifiers working on a probabilistic model [37]. GNB is a variation of Naive Bayes classifiers, which are based on the Bayes theorem. The classifier is trained with various functions implemented for the distribution estimation of data, and the easiest implemented is the Gaussian function. Meanwhile, it would need estimation from the training data, standard deviation, and mean. In GNB, the purpose of the probability function is to predict the Gaussian function will provide a likelihood estimate of the new input value by simply swapping out the parameters with the variable's fresh values entered.

9) Stochastic Gradient Descent (SGD)

SGD is a technique used to iteratively enhance the smoothness properties of an objective function to find optimal parameter values [38]. SGD employs an estimated gradient produced from a random subset of the data, as opposed to computing the gradient from the complete dataset. It is an effective and useful optimization technique because of its stochastic approximation for minimizing the function of cost. When learning convex loss function linear classifiers discriminatively, SGD is used, and each training instance updates the coefficients. For high-dimensional data, it is an affordable solution that can hasten convergence.

10) Feature engineering using CNN

In this study, feature extraction is made using the CNN described in [39] to predict students' academic performance. It contains four main layers: the convolutional layer, the flattening layer, an embedding layer, and a max-pooling layer. It uses a 1D convolutional layer. The dataset has twenty features embedded with an embedding layer of 11 inputs, 300 outputs, and a vocabulary size 20,000. The subsequent one is a 1D convolutional layer of size 2×2, along with 5,000 filters and Rectified Linear Unit (ReLU) activation. After that, a size 2×2 max-pooling layer was used for soliciting the essential features. Finally, the flattened layer converts the output into a 1D array so that it works correctly with ML models. The hyperparameter details of the modified CNN framework is shared in Table 4.

The embedding layer is applied to transform the training set in the format required for input. Dataset A: contains a set of tuples (f_s_i, t_c_i), where i is the tuple index, the target class column is denoted by t_c, and the feature set by f_s.

$$EL = embedding\_layer(V_s, O_s, I)$$

Table 4. CNN hyperparameters used in SuccessNet framework for feature engineering

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Epochs	100
Loss Function	Categorical Cross-Entropy
Validation Split	20%
Activation Functions	ReLU (hidden), Softmax (output)
Random Seed	42 (for reproducibility)

$$EO_s = EL(f\_s)$$

In this equation, EO_s represents the output of the embedding layer and takes up the function of the convolutional layer's input. EL stands for parameters that specify the input length for the embedding layer, output dimensions, and vocabulary sizes compared. This research will consider the model input dimension, which is obtained through the vocabulary dimension, $V_s = 20{,}000$. It implies that the inputs in the model during training range between 0 and 20,000. The output dimension parameter of the embedding layer is set to 250. This means that the data at post-embedding layer processing will have a dimension of 250. The input length is the final parameter of the embedding layer. This is represented by the variable I. The variable I equals the number of available features in the dataset. In this case, I = 20. The embedding layer modifies the incoming data, producing output that the CNN model can use to process the data further. The dimensions of the embedding layer's output are provided as:

$$EO_s = (None, 11, 250) \tag{1}$$

$$1D - Convs = CNN(F, Ks, AF) \leftarrow EO_s \tag{2}$$

The 1D convolutional layers' output is displayed as $1D - Convs$., taken from the output of the embedding layer. Feature extraction applies 5,000 filters (F = 5,000). where the size of the kernel used is Ks = 2×2. The ReLU Activation Function (AF) applies the following operation to the elements of the $1D - Convs$ output matrix, that is, put all the non-positive figures as 0, leaving others unchanged:

$$f(x) = \max(0, E) \tag{3}$$

Max-pooling extracts the essential features from CNN. Here, we have taken the feature map pool size 2×2. The pooling window size and stride are fixed at 2, i.e., ($Ps = 2$).

$$Cf = Famp = \lfloor (1 - P_s) / S \rfloor + 1 \tag{4}$$

A flattened layer converts 3D data into 1D and, therefore can work better with ML algorithms. This is a significant conversion because most algorithms are known to perform best with 1D data.

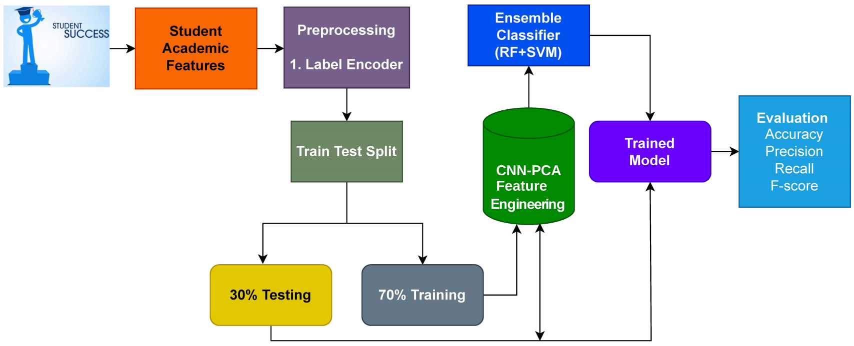


Fig. 1. Proposed system (RF+SVM) work flow diagram.

Algorithm 1 Ensembling RF and SVM.

Input: input data $(x, y)_{i=1}^{N}$

M_{RF} = Trained RF

M_{SVM} = Trained SVM

for i = 1 to M do

if $M_{RF} \neq 0$ & $M_{SVM} \neq 0$ & $training\_set \neq 0$ then

$P_{RF_1} = M_{RF_1}.probability(class1)$

$P_{RF_2} = M_{RF_2}.probability(class2)$

$P_{RF_3} = M_{RF_3}.probability(class3)$

$P_{SVM_1} = M_{SVM_1}.probability(class1)$

$P_{SVM_2} = M_{SVM_2}.probability(class2)$

$P_{SVM_3} = M_{SVM_3}.probability(class3)$

Decision function =

$\max(\frac{1}{n}\sum_{classifier}(Avg(P_{RF_1}, P_{SVM_1}), Avg(P_{RF_2}, P_{SVM_2}), Avg(P_{RF_3}, P_{SVM_3}))$

end if

return final label p

end for

C. Proposed Methodology

For this reason, most studies have paid attention to using group ML models to improve efficiency. The demonstration of ensemble classifiers is usually better than that of individual models. Because of this success remark, an ensemble model will be utilized in the current study to predict student academic success. Fig. 1 presents the flow of the prediction of student academic success. SVM and RF are two ML methods, are merged in the proposed methodology to perform experiments by employing a particular dataset regarding student academic performance. Experiments with the student dataset have been executed with the model in this research work, considering two scenarios. In the first scenario, to predict academic performance, the student dataset's 20 attributes are used. Now, the CNN model takes in features from this dataset. These attributes are given to teach the model how to categorize the students' grades into G1, G2, and G3 categories (High, Medium, and Low Performance). Data are divided into 70% to 30% percent for testing and training. Performance metrics such as recall, accuracy, precision, and F1-Score are used to evaluate the model. This paper proposes an ensemble of RF and SVM using the soft voting criterion. The result with the highest probability is selected as the conclusion in gentle voting. Algorithm 1 illustrates how the suggested ensemble model functions. The probability values given for each instance are then soft-voted using RF and SVM. While equal weighting (i.e., 0.5 each) was used in this study to maintain fairness and simplicity, we also performed initial experiments with different weight distributions based on validation accuracy (e.g., RF: 0.6, SVM: 0.4) and found negligible performance gain over equal weights. Hence, uniform weights were adopted for reproducibility and generalizability.

D. Mathematical Modeling of SuccessNet

1) CNN for feature extraction

Let $X \in \mathbb{R}^{n \times d}$ denote the input data matrix, where n is the number of samples and d is the number of features. The CNN operates on this input data to extract higher-dimensional features.

a) Convolutional layer

For a convolutional layer, let W_c be the convolutional filter of size $k \times k$, and b_c be the bias term. The output of the convolutional operation O_c is given by:

$$O_c = f(W_c * X + b_c)$$

where $*$ denotes the convolution operation and f is the activation function, typically a ReLU:

$$f(z) = \max(0, z)$$

b) Pooling layer

The pooling layer downsamples the feature map. If we use max-pooling with a pool size of $p \times p$, the output O_p is:

$$O_p = \max(O_c[i : i + p - 1, j : j + p - 1])$$

where i and j are the sliding window indices over the feature map.

c) Fully connected layer

The output from the convolutional and pooling layers is flattened and fed into a fully connected layer. Let W_{fc} and b_{fc} be the weights and bias of the fully connected layer, respectively. The output O_{fc} is:

$$O_{fc} = f(W_{fc} \cdot O_p + b_{fc})$$

2) PCA for dimensionality reduction

PCA is applied to the features extracted by the CNN to reduce dimensionality. Let ($Z \in \mathbb{R}^{n \times m}$) be the matrix of extracted features where (m) is the number of features after flattening. The PCA transformation Z_{pca} is given by:

$$Z_{pca} = Z \cdot W_{pca}$$

where $W_{pca} \in \mathbb{R}^{m \times k}$ is the matrix of principal components and k is the number of principal components selected.

3) RF and SVM ensemble model

a) RF

Let T denote the number of trees in the random forest. Each tree t in the forest is trained on a bootstrap sample of Z_{pca}. The prediction of the t-th tree for a sample z is $h_t(z)$. The RF prediction H_{rf} is the majority vote (or average for regression):

$$H_{rf}(z) = \text{mode}(h_1(z), h_2(z), ..., h_T(z))$$

b) SVM

The SVM classifier constructs a hyperplane that maximizes the margin between classes. For a linear SVM, the decision function f_{svm} is:

$$f_{svm}(z) = W_{svm} \cdot z + b_{svm}$$

where W_{svm} and b_{svm} m are the weight vector and bias term, respectively.

c) Ensemble model prediction

The final prediction H is obtained by combining the predictions of RF and SVM. Let α be the weighting factor. The ensemble prediction is:

$$H(z) = \alpha \cdot H_{rf}(Z) + (1-\alpha) \cdot f_{svm}(z)$$

where $\alpha \in [0, 1]$ balances the contribution of RF and SVM.
Summary of the Mathematical Model

CNN Feature Extraction:

$$O_{fc} = f(W_{fc} \cdot \text{flatten}(O_p) + b_{fc})$$

PCA Transformation:

$$Z_{pca} = Z \cdot W_{pca}$$

RF Prediction:

$$H_{rf}(z) = \text{mode}(h_1(z), h_2(z), ..., h_T(z))$$

SVM Prediction:

$$f_{svm}(z) = W_{svm} \cdot z + b_{svm}$$

Ensemble Prediction:

$$H(z) = \alpha \cdot H_{rf}(z) + (1-\alpha) \cdot f_{svm}(z)$$

This mathematical modeling outlines the architecture and functioning of the CNN feature extraction with the RF+SVM ensemble model for predicting student academic success.

The original attributes (as shown in Table 3) consist of 20 manually recorded features, such as academic scores (CGPA, CW1, CW2, ESE), behavioral metrics (video plays, pauses, likes), academic risk indicators (AtRisk, HighRisk), and administrative records (PlagiarismHistory, Probation, TermExceeded). These features are tabular, domain-specific, and relatively shallow in representational depth, meaning they directly describe observed variables without learning underlying relationships.

In contrast, the CNN-generated features are learned representations derived by reshaping the original tabular data into a 2D matrix and feeding it through a customized 1D CNN with filters designed to detect local patterns and co-occurrence structures across related academic variables. For instance, the CNN captures interactions between academic performance trends (CW1, CW2, ESE) and student engagement metrics (Played, Likes, Segment) that are not explicitly encoded in the original features. After convolution and flattening, the CNN yields 512 high-dimensional features per instance. To reduce redundancy and retain only the most informative aspects, we apply PCA, which reduces the 512 CNN features to 326 principal components by preserving 95% of variance. This step enhances generalization by removing noise and emphasizing orthogonal dimensions of variance that contribute most to class separability. This transformation substantially improves the model's ability to detect subtle correlations and patterns in student behavior and academic trajectory, which are critical for accurate academic performance prediction.

E. Evaluation Metrics

The suggested system's accuracy in forecasting student performance can be measured with the help of four basic units, which include: True Positive (TP), TN stands for True Negative, FP for False Positive, and FN for False Negative. Furthermore, the following are employed. The system's accuracy is determined by how well it can forecast students' academic performance based on the target data set. It is measured as the proportion of real negatives and true positives, calculated over all cases observed, by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision is the genuine positive metric, and it may be computed as

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Another name for the recall is sensitivity. Recall can be computed using the true positive percentage as

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

To get a single statistic, the F-Score is the harmonic mean of recall and precision. It may be provided by

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

IV. RESULT AND DISCUSSION

A. Experiment Set up

Different types of experimentation for student performance prediction are done using a Dell PowerEdge T720 with a 2 GB GPU, two Intel Xeon processors with eight cores each operating at 2.4 GHz, and 16 GB DDR4 RAM. A Jupyter Notebook with Python is used to execute the experiments, where all ML models are developed with the help of the sci-kit-learn library in Python. The output of the ensemble and base models is shown in this section. The experiments are carried out utilizing the initial 20 features, and as a method of feature engineering, the features are retrieved using CNN. The resulting findings will be compared with those of other cutting-edge models.

B. Results of the ML Models

Using CNN's feature engineering, we extracted unique features and trained ML models on them. A total of nine ML models is considered for evaluating the system's effectiveness. The way these models performed with the original features and the features processed through CNN is shown in Table 5 and in Fig. 2.

Table 5. ML models results comparison

Model	Accuracy with original attributes (%)	Accuracy with convoluted attributes (%)
DT	82.09	85.73
ETC	84.78	88.55
GBM	80.63	82.63
GNB	79.66	84.34
KNN	76.99	81.76
LR	83.45	86.63
RF	86.53	88.32
SGD	79.84	85.38
SVM	83.98	87.41

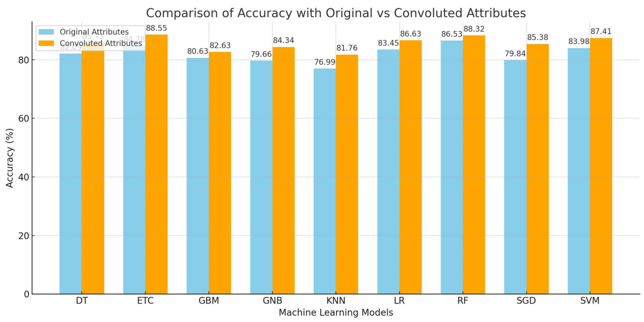


Fig. 2. ML models results comparison using both features.

Table 5 shows the accuracy results for nine various ML algorithms using the original features and those extracted through CNN. Specifically, the Extra Trees Classifier gave the best percentage of accuracy with convoluted features at 88.55%, which is improved and quite significant when compared to its original feature accuracy of 84.78%. This means that ETC benefits greatly from the feature-extraction process in such a way that its prediction capability is enhanced. Next, after ETC, the RF model also performed excellently, giving an accuracy of 88.32% using convoluted features compared to 86.53% using original features. This proves the robustness of the RF model and that the enhanced performance was due to more refined features. SVM demonstrated quite a good improvement, too, as its accuracy increased from 83.98% using the original feature to 87.41% when the features were convoluted. This may point out the utility of making the classification accuracy of SVM increase through feature extraction. We also observe a great increase in other classifiers, Logistic Regression (LR), from 84.23% with original features to 86.63% with convoluted features. The other models, like DT, GBM, GNB, KNN, and SGD, also improved using convoluted features, which overall proves that feature extraction through CNN improves the performance of most ML models.

C. Proposed Ensemble Model Results on the Original Features

The results of applying the ensemble models on the original features are displayed in the Table 6. The following four ML models serve as the foundation for the ensemble: SVM, RFs, Logistic Regression, and Extra Trees Classifier. All these base classifiers are selected because each works well as a standalone classifier. The classification results of all the Table 6 list the models that were applied to the original features.

Table 6. Complete features set results

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
ETC+LR	88.77	89.36	89.71	89.49
ETC+SVM	92.77	91.49	93.67	92.57
RF+ETC	91.37	91.27	93.42	92.35
RF+LR	87.9	88.56	86.76	87.66
RF+SVM	93.58	93.51	96.77	94.63
SVM+LR	89.62	90.86	95.01	93.93

An experimental result of an ensemble learning model is shown in Table 6 where the number of features is equal to the original feature set. The best-observed classification performance is from RF and SVM combination, which achieved 93.55% accuracy, 93.48% precision, 96.74% recall, and 94.6% F-Score. The combination shows better performance in all aspects, indicating a well-built model. The next best was the ensemble of ETC and SVM, achieving a 92.74% accuracy, 91.46% precision, 93.64% recall, and 92.54 F-Score. The RF, ETC combination also did quite well, producing a 91.34% accuracy, 91.24% precision, 93.39% recall, and a 92.32% F-Score. The SVM and LR ensemble achieved a high accuracy of 89.59%, with respective precisions and recalls of 90.83% and 94.98%, accumulating an F-Score of 93.9%. Next in line was the ETC and LR combination, which scored an accuracy of 88.74% and provided precisions of 89.33% and recalls of 89.68%, accumulating an F-measure of 89.46%. The RF and LR ensemble demonstrated the best performance, yielding an F-Score of 87.63%, accuracy of 87.87%, precision of 88.53%, and recall of 86.73%. The set with the most accurate and reliable predictions was RF+SVM.

D. Results of All Ensemble Learning Models with Convoluted Features

The experimental results of ensemble learning models with a complex feature set are shown in Table 7. The combination of RF and SVM achieved the highest performance, with an outstanding accuracy of 99.35% and exceptional precision, recall, and F-Score all at 99.87%. This indicates an almost perfect performance, showcasing the robustness and reliability of this ensemble. Following this, the ensemble of Extra Trees Classifier (ETC) and SVM also demonstrated excellent performance with an accuracy of 98.87%, precision of 98.66%, recall of 98.29%, and an F-Score of 98.46%. The RF, ETC combination showed strong results, achieving an accuracy of 98.38%, precision of 98.22%, recall of 99.41%, and an F-Score of 99.33. The RF and Logistic Regression (LR) ensemble achieved a notable accuracy of 94.63%, with a precision of 95.31%, recall of 98.49%, and an F-Score of 96.4%. Lastly, the ETC and LR ensemble followed with 93.42% accuracy, 94.84% precision, 98.61% recall, and 96.74% F-Score. Overall, the RF+SVM ensemble is the most accurate and reliable model, demonstrating superior performance across all metrics. The confusion matrix with FP = 2, TP = 263, TN = 61, and FN = 0 is shown in Table 8 respectively. The ensemble results comparison using both features are shared in Fig. 3. Fig. 4 shows the complete ROC and AUC curves. The t-test analysis between the ETC+SVM and RF+SVM models yielded the following results: t-statistic: 6.48, p-value: 0.00064.

Table 7. Results of ensemble models with convolutional features

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
ETC+LR	93.42	94.84	98.61	96.74
ETC+SVM	98.87	98.66	98.29	98.46
RF+ETC	98.38	98.22	99.41	99.33
RF+LR	94.63	95.31	98.49	96.4
RF+SVM	99.35	99.87	99.87	99.87

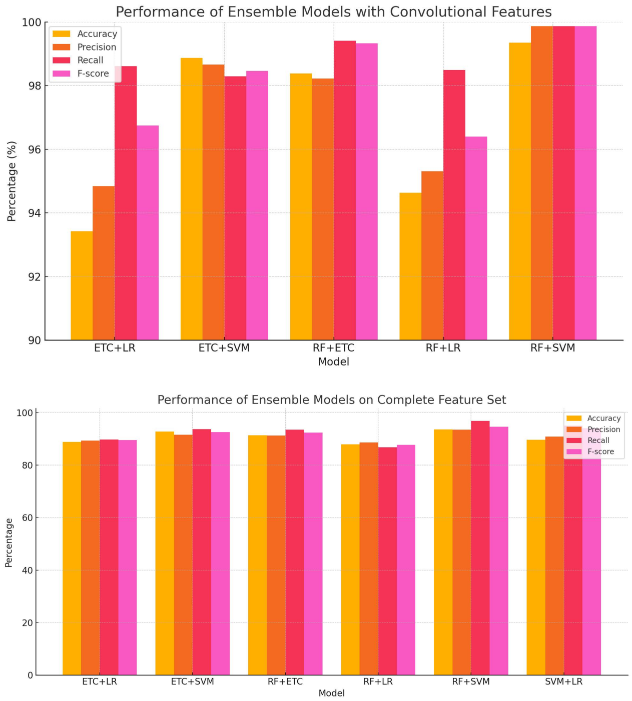
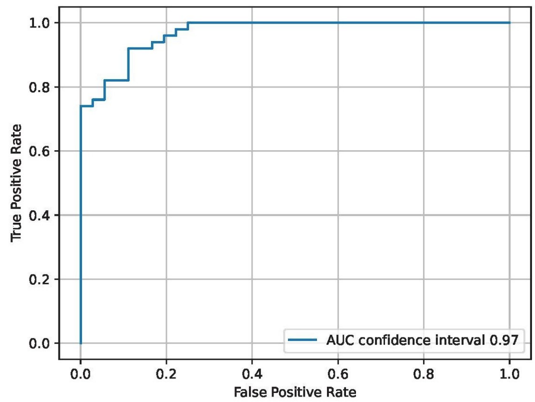


Fig. 3. Ensemble results comparison using both features.

Table 8. Confusion matrix of proposed model

	Predicted Pass	Predicted Fail
Actual Pass	263	0
Actual Fail	2	61



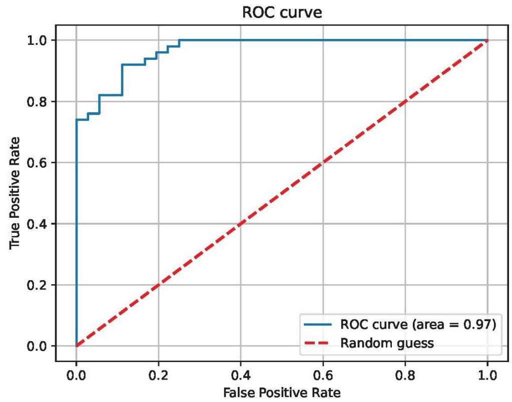


Fig. 4. AUC-ROC curve explanation.

These values indicate a statistically significant difference between the two models' performance metrics at conventional significance levels (e.g., $p < 0.05$). Thus, the RF+SVM model performs significantly better than the ETC+SVM model.

E. K-Fold Results of Cross Validation

The k-fold validation technique is utilized in this research to check the significance of the proposed framework on all subsets of the dataset. Table 9 shows the 10-fold cross-validation result. The accuracy results of all subsets range from 0.969–0.986 which shows the stability of the proposed model.

Table 9. Cross-validation result of the proposed framework

Fold No.	Accuracy	Precision	Recall	F1-Score
Fold-01	0.969	0.972	0.971	0.972
Fold-02	0.971	0.973	0.972	0.973
Fold-03	0.973	0.974	0.973	0.974
Fold-04	0.975	0.976	0.986	0.975
Fold-05	0.976	0.976	0.975	0.975
Fold-06	0.986	0.976	0.976	0.975
Fold-07	0.972	0.976	0.973	0.974
Fold-08	0.974	0.975	0.974	0.975
Fold-09	0.974	0.974	0.975	0.975
Fold-10	0.976	0.976	0.976	0.976

A summary of the performance metrics of the model for 10-fold cross-validation is displayed in Table 9. The table presents accuracy, recall, precision, and F-Score for every fold. For the best accuracy, it can be noted to be 0.986 for Fold-06. Besides, the precision is also very high at this fold: 0.976, and the recall is the same: 0.976. Hence, the F-Score totaled 0.975. It meant that the model was very reliable and consistent in this fold, giving out high performance in all the measures. This was followed very closely by Fold-05 and Fold-10. These are where similar levels of accuracy to 0.976 have been recorded, along with precision, recall, and F-Score. These results suggest that across the folds of these two performances by the model, decent balanced precision and good recall is explaining the high F-Score. Fold-04 also performs well, with an accuracy of 0.975 and the highest score in recall of 0.986. In it, the precision reaches 0.976, and its F-Score is evaluated at 0.975, showing how the model performed relatively well in pinpointing relevant instances correctly, given a high recall. The accuracy measures of other folds range from 0.969 to 0.974, with their corresponding precision, recall, and F-Scores distributing around values generally as high as those ending between 0.972 and 0.975. Such consistent performance across all folds demonstrates proving the designed model is dependable and stable, while at the same time, minor variation in metrics indicates stable and dependable results. The classifier consequently shows high accuracy, precision, recall, and F-Score across almost all six folds, being highest in Fold-06. These results reasonably well reflect good predictivity of the model in a balanced performance in classifying the instances.

F. Ablation Study

We conducted an ablation study to validate each component's effectiveness in the SuccessNet framework. Using only CNN features achieved 91.8% accuracy, while integrating PCA improved it to 94.7% by reducing feature redundancy. Applying RF and SVM individually on CNN+PCA features yielded 96.8% and 97.1% accuracy, respectively. The complete ensemble with soft voting (RF+SVM) achieved the highest accuracy of 99.35%, confirming that each component—CNN, PCA, and ensemble—contributes significantly to performance.

G. Novelty of the Research in Student Success Prediction

The proposed research introduces several novel contributions to the field of student success prediction. The use of CNN for extracting features from educational data is a significant innovation. CNNs are typically employed in image processing tasks, but their application to student performance prediction is novel. By leveraging CNNs, the model can automatically learn and extract high-level features from raw input data, capturing intricate patterns and relationships that traditional methods might miss. Incorporating PCA for dimensionality reduction in conjunction with CNN features is another novel aspect. This step ensures that the model handles the high-dimensional data effectively, reducing computational complexity while preserving essential information. This hybrid approach of CNN feature extraction followed by PCA is not commonly explored in EDM. The combination of RF and SVM in an ensemble learning framework is a key innovation. While RF and SVM have been used individually for classification tasks, their combined use, particularly in the context of student success prediction, enhances the model's robustness and accuracy. The ensemble approach leverages the strengths of both classifiers: RF's ability to handle noisy and non-linear data, and SVM's proficiency in creating optimal decision boundaries. The research presents a unique fusion strategy for integrating CNN-extracted features and PCA-transformed data with the RF+SVM ensemble model. This method ensures that the complementary strengths of different techniques are utilized effectively, leading to improved predictive performance. The careful design of the fusion process, including the optimal weighting of RF and SVM contributions, is a novel aspect that adds to the model's efficacy. The application of this advanced modeling technique to diverse types of educational data, including both structured (e.g., grades, attendance) and unstructured data (e.g., text from assignments), showcases the model's versatility. The ability to adapt and perform well across different data formats and educational contexts is a significant advancement in the field. By combining the interpretability of RF (through feature importance scores) and the clear decision boundaries provided by SVM, the model not only achieves high accuracy but also offers insights into the key factors influencing student success. This dual benefit of accuracy and interpretability is a novel contribution that addresses the need for transparent and actionable predictive models in education. In summary, the research presents a pioneering approach to student success prediction by integrating CNN for feature extraction, PCA for dimensionality reduction, and an RF+SVM ensemble for robust classification. This combination of techniques, along with the novel fusion strategy and applicability to diverse educational data, sets this research apart as a significant advancement in the field of EDM and student performance prediction.

H. Significance of the Proposed Model

Combining CNN for feature extraction with a RF and SVM ensemble model for classification enhances the prediction of student academic performance due to several key factors. CNNs excel at automatically learning and extracting high-level features from raw data, identifying complex patterns and structures through a hierarchical feature extraction process. This allows CNNs to capture nuanced details in continuous numerical and categorical data. The RF and SVM classifiers offer robust and complementary strengths: RF reduces overfitting by averaging multiple decision trees, while SVM handles high-dimensional spaces with clear decision boundaries. This diversity improves generalization, reduces prediction variance, and ensures stability. RF's feature importance scores and SVM's decision boundaries enhance interpretability, making it easier to understand which features influence aca-demic performance. Additionally, CNNs adapt to various data types, enabling the model to process both structured (numerical grades, attendance) and unstructured data (text). The combination of CNNs with RF and SVM scales efficiently to large datasets and making it ideal for educational settings. This synergy captures complex patterns, boosts generalization, and provides clear interpretability, resulting in superior performance for predicting student academic outcomes.

I. Limitations of the Proposed Model

The proposed model is based on the fusion of PCA-extracted convoluted features and RF+SVM ensemble learning model may suffer from some limitations. The proposed framework's performance is heavily dependent on the quantity and quality of available data, with insufficient or biased data potentially leading to inaccurate predictions. Additionally, a lack of diverse data representing different demographics and academic backgrounds can limit the model's generalizability. The model's complexity, combining CNN for feature extraction with RF and SVM ensemble learning is computationally intensive and requires substantial resources. Adapting the model to different types of educational data, such as online learning platforms or traditional classroom settings, might require significant modifications. In summary, while the proposed model leverages advanced techniques for feature extraction and classification, its complexity, potential for overfitting, computational demands, and challenges in feature integration and interpretation are notable limitations for robust and scalable academic success prediction.

V. Conclusion

This research work proposes a novel framework called

SuccessNet to predict the student's academic performance using CNN-generated and PCA extracted significant features with RF and SVM soft ensemble learning model. The experimentation is carried out in two steps. The first step makes use of the original dataset features with all individual machine-learning models and ensemble models. In the second step, the CNN and PCA extracted significant features are utilized with the same individual machine-learning models and ensemble models. The best performance is given by the ensemble of RF and SVM models with 99.35% accuracy and 99.87% recall, precision, and F-Score. This approach empirically outperforms the present state-of-the-art studies. Future developments could be related to an ensemble by examining DL and ML models to forecast academic success. This method could be generalized for other datasets.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

K. A. performed formal analysis, reviewed, and edited the original manuscript, acquired funding, performed visualization, and designed methodology. M. U. conceived the idea, performed data curation, designed methodology, and wrote the original manuscript. All authors had approved the final version.

FUNDING

This project was funded by the Deanship of Scientific Research at Prince Sattam bin Abdulaziz University award number 2023/01/24320.

REFERENCES

[1] J. Shailaja and R. Sridaran, "Taxonomy of e-learning challenges and an insight to blended learning," in *Proc. 2014 IEEE International Conference on Intelligent Computing Applications (ICICA)*, 2014, pp. 1–6. doi: 10.1109/ICICA.2014.70

[2] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat, "Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic," *Educ. Inf. Technol.*, vol. 25, pp. 5261–5280, May 2020. https://doi.org/10.1007/s10639-020-10219-y

[3] A. A. M. Al-araibi, M. N. B. Mahrin, and R. C. M. Yusoff, "Technological aspect factors of E-learning readiness in higher education institutions: Delphi technique," *Educ. Inf. Technol.*, vol. 24, pp. 567–590, Aug. 2019. https://doi.org/10.1007/s10639-018-9780-9

[4] P. Vazan, D. Janikova, P. Tanuska, M. Kebisek and Z. Cervenanska, "Using data mining methods for manufacturing process control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6178–6183, Jul. 2017. https://doi.org/10.1016/j.ifacol.2017.08.986

[5] A. Ishaq, S. Sadiq, M. Umer *et al.*, "Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques," *IEEE Access*, vol. 9, pp. 39707–39716, Mac. 2021. doi: 10.1109/ACCESS.2021.3064084

[6] C. Shah, Q. Du, and Y. Xu, "Enhanced TabNet: Attentive interpretable tabular learning for hyperspectral image classification," *Remote Sens.*, vol. 14, no. 3, 716, Feb. 2022. https://doi.org/10.3390/rs14030716

[7] L. Jackson, *Get the 411: Laptops and Tablets in the Classroom*, 2013.

[8] W. F. W. Yaacob, S. A. M. Nasir, W. F. W. Yaacob, and N. M. Sobri, "Supervised data mining approach for predicting student performance," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 3, pp. 1584–1592, Dec. 2019. http://doi.org/10.11591/ijeecs.v16.i3.pp1584-1592

[9] A. Zohair and L. Mahmoud, "Prediction of student's performance by modelling small dataset size," *Int. J. Educ. Technol. High. Educ.*, vol. 16, no. 1, pp. 1–18, Aug. 2019. https://doi.org/10.1186/s41239-019-0160-3

[10] M. N. Hasnine, G. Akcapinar, B. Flanagan *et al.*, "Towards final scores prediction over clickstream using machine learning methods," in *Proc. 26th Int. Conf. Comput. Educ.*, 2018, pp. 399–404.

[11] N. A. Kuadey, C. Ankora, F. Tahiru *et al.*, "Using machine learning algorithms to examine the impact of technostress creators on student learning burnout and perceived academic performance," *Int. J. Inf. Technol.*, vol. 16, no. 4, pp. 2467–2482, Dec. 2024. https://doi.org/10.1007/s41870-023-01655-3

[12] M. M. Hussain, S. Akbar, S. A. Hassan, M. W. Aziz, and F. Urooj, "Prediction of student's academic performance through data mining approach," *J. Inform. Web Eng.*, vol. 3, no. 1, pp. 241–251, Feb. 2024. https://doi.org/10.33093/jiwe.2024.3.1.16

[13] P. Umamaheswari, M. Vanitha, P. V. Devi, J. G. Theporal, and B. R. Basha, "Student success prediction using a novel machine learning approach based on modified SVM," *Multidiscip. Sci. J.*, vol. 6, 2024ss0110, Dec. 2023. https://10.31893/multiscience.2024ss0103

[14] K. S. Narayanan and A. Kumaravel, "A novel Chaotic Optimized Boost Long Short-Term Memory (COB-LSTM) model for students academic performance prediction in educational sectors," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 2s, pp. 519–528, Oct. 2024. https://ijisae.org/index.php/IJISAE/article/view/3652

[15] S. Sarwat, N. Ullah, S. Sadiq *et al.*, "Predicting students' academic performance with conditional generative adversarial network and deep SVM," *Sensors*, vol. 22, no. 13, 4834, Jun. 2022. https://doi.org/10.3390/s22134834

[16] F. J. Kaunang and R. Rotikan, "Students' academic performance prediction using data mining," in *Proc. 2018 Third International Conference on Informatics and Computing (ICIC)*, 2018, pp. 1–5. doi: 10.1109/IAC.2018.8780547

[17] C. F. Rodriguez-Hernandez, M. Musso, E. Kyndt, and E. Cascallar, "Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation," *Comput. Educ. Artif. Intell.*, vol. 2, 100018, Apr. 2021. https://doi.org/10.1016/j.caeai.2021.100018

[18] N. Aslam, I. Khan, L. Alamri, and R. Almuslim, "An improved early student's academic performance prediction using deep learning," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 12, pp. 108–122, Jun. 2021. https://doi.org/10.3991/ijet.v16i12.20699

[19] A. Rivas, A. Gonzalez-Briones, G. Hernandez, J. Prieto, and P. Chamoso, "Artificial neural network analysis of the academic performance of students in virtual learning environments," *Neurocomputing*, vol. 423, pp. 713–720, Jun. 2021. https://doi.org/10.1016/j.neucom.2020.02.125

[20] S. Poudyal, M. J. Mohammadi-Aragh, and J. E. Ball, "Prediction of student academic performance using a hybrid 2D CNN model," *Electronics*, vol. 11, no. 7, p. 1005, Mar. 2022. https://doi.org/10.3390/electronics11071005

[21] M. Ya˘gcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learn. Environ.*, vol. 9, no. 1, p. 11, Mar. 2022. https://doi.org/10.1186/s40561-022-00192-z

[22] N. N. Hamadneh, S. Atawneh, W. A. Khan, K. A. Almejalli, and A. Alhomoud, "Using artificial intelligence to predict students' academic performance in blended learning," *Sustainability*, vol. 14, no. 18, 11642, Sep. 2022. https://doi.org/10.3390/su141811642

[23] C. Liu, H. Wang, and Z. Yuan, "A method for predicting the academic performances of college students based on education system data," *Mathematics*, vol. 10, no. 20, p. 3737, Oct. 2022. https://doi.org/10.3390/math10203737

[24] A. D. Ali and W. K. Hanna, "Predicting students' achievement in a hybrid environment through self-regulated learning, log data, and course engagement: A data mining approach," *J. Educ. Comput. Res.*, vol. 60, no. 4, pp. 960–985, Dec. 2022. https://doi.org/10.1177/07356331211056178

[25] S. Chen and Y. Ding, "A machine learning approach to predicting academic performance in Pennsylvania's schools," *Soc. Sci.*, vol. 12, no. 3, p. 118, Feb. 2023. https://doi.org/10.3390/socsci12030118

[26] M. A. Baig, S. A. Shaikh, K. K. Khatri *et al.*, "Prediction of students performance level using integrated approach of ML algorithms," *Int. J. Emerg. Technol. Learn.*, vol. 18, no. 1, pp. 216–234, Jan. 2023. https://doi.org/10.3991/ijet.v18i01.35339

[27] Y. T. Badal and R. K. Sungkur, "Predictive modelling and analytics of students' grades using machine learning algorithms," *Educ. Inf. Technol.*, vol. 28, no. 3, pp. 3027–3057, Sep. 2023. https://doi.org/10.1007/s10639-022-11299-8

[28] R. Hasan, S. Palaniappan, S. Mahmood, A. Abbas, and K. U. Sarker, "Dataset of students' performance using student information system, Moodle and the mobile application 'eDify'," *Data*, vol. 6, no. 11, p. 110, Oct. 2021. https://doi.org/10.3390/data6110110

[29] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996. https://doi.org/10.1007/BF00058655

[30] M. Manzoor, M. Umer, S. Sadiq *et al.*, "RFCNN: traffic accident severity prediction based on decision level fusion of machine and deep

learning model," *IEEE Access*, vol. 9, pp. 128359–128371, Sep. 2021. 10.1109/ACCESS.2021.3112546

[31] A. Juna, M. Umer, S. Sadiq *et al.*, "Water quality prediction using KNN imputer and multilayer perceptron," *Water*, vol. 14, no. 17, p. 2592, Aug. 2022. https://doi.org/10.3390/w14172592

[32] E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 9, pp. 3669–3692, Sep. 2019. https://doi.org/10.1007/s12652-018-1093-8

[33] C. Khammassi and S. Krichen, "A NSGA2-LR wrapper approach for feature selection in network intrusion detection," *Comput. Netw.*, vol. 172, 107183, May 2020. https://doi.org/10.1016/j.comnet.2020.107183

[34] I. Ashraf, M. Narra, M. Umer *et al.*, "A deep learning-based smart framework for Cyber-Physical and satellite system security threats detection," *Electronics*, vol. 11, no. 4, p. 667, Feb. 2022. https://doi.org/10.3390/electronics11040667

[35] M. Umer, S. Sadiq, M. Nappi *et al.*, "ETCNN: extra tree and convolutional neural network-based ensemble model for COVID-19 tweets sentiment classification," *Pattern Recognit. Lett.*, vol. 164, pp. 224–231, Dec. 2022. https://doi.org/10.1016/j.patrec.2022.11.012

[36] F. Rustam, A. A. Reshi, W. Aljedaani *et al.*, "Vector mosquito image classification using novel RIFS feature selection and machine learning models for disease epidemiology," *Saudi J. Biol. Sci.*, vol. 29, no. 1, pp. 583–594, Jan. 2022. https://doi.org/10.1016/j.sjbs.2021.09.021

[37] R. Majeed, N. A. Abdullah, M. F. Mushtaq, M. Umer, and M. Nappi, "Intelligent cyber-security system for IoT-aided drones using voting classifier," *Electronics*, vol. 10, no. 23, 2926, Nov. 2021. https://doi.org/10.3390/electronics10232926

[38] M. Umer, S. Sadiq, M. M. S. Missen *et al.*, "Scientific papers citation analysis using textual features and SMOTE resampling techniques," *Pattern Recognit. Lett.*, vol. 150, pp. 250–257, Oct. 2021. https://doi.org/10.1016/j.patrec.2021.07.009

[39] A. Hameed, M. Umer, U. Hafeez *et al.*, "Skin lesion classification in dermoscopic images using stacked convolutional neural network," *J. Ambient Intell. Humaniz. Comput*, vol. 14, no. 4, pp. 3551–3565, Sep. 2021. https://doi.org/10.1007/s12652-021-03485-2