digital

Article

A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance

Kuburat Oyeranti Adefemi * and Murimo Bethel Mutanga

Department of Information and Communication Technology, Mangosuthu University of Technology, Umlazi, Durban 4026, South Africa; mutangamb@mut.ac.za
* Correspondence: adefemi.kuburat@mut.ac.za or adefemioyeranti@gmail.com

Abstract: The rapid increase in educational data from diverse sources such as learning management systems and assessment records necessitates the application of advanced analytical techniques to identify at-risk students and address persistent issues like dropout rates and academic underperformance. However, many existing models struggle with generalizability and fail to effectively manage data challenges such as class imbalance and missing data, leading to suboptimal predictive performance. This study proposes a hybrid deep learning model combining convolutional neural networks (CNN) and long short-term memory (LSTM) networks to improve the accuracy of student academic performance prediction and enable timely educational interventions. To improve the performance of the model, we incorporate feature selection techniques and optimization strategies to enhance reliability. We also address common preprocessing challenges such as missing data and data imbalance. The proposed model was evaluated on two benchmark datasets to ensure model generalization capability. The hybrid model achieved predictive accuracies of 98.93% and 98.82% on the two datasets, respectively, outperforming traditional machine learning models and standalone deep learning approaches across key performance metrics including accuracy, precision, recall, and F-score.

Keywords: educational data mining; deep learning; student performance prediction

check for updates

Academic Editor: Yannis Manolopoulos

Received: 30 January 2025
Revised: 22 April 2025
Accepted: 18 May 2025
Published: 21 May 2025

Citation: Adefemi, K.O.; Mutanga, M.B. A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance. *Digital* **2025**, *5*, 16. https://doi.org/10.3390/digital5020016

1. Introduction

Education plays a major role in personal development by acquiring and impacting knowledge, critical thinking, and personal transformation. Predicting student performance is crucial in academic institutions for supporting strategic decision-making and developing targeted interventions. Furthermore, with the rapid increase in educational data, higher education institutions increasingly prioritize strategies to monitor and enhance student academic performance.

To support this effort, the growing field of educational data mining (EDM) has emerged as a critical tool to enhance teaching strategies, support decision-making, and improve student outcomes by studying the massive data generated by educational institutions [1,2]. EDM uses various techniques such as clustering, classification, and predictive modeling to analyze educational data, which includes student demographics, academic performance, learning behaviors, etc. [3,4]. Prior studies have employed machine learning algorithms such as neural networks, decision trees, naïve bayes, k-nearest neighbor, random forest, logistic regression, support vector machines, and ensemble classifiers to predict student academic outcomes. For instance, the work by [5] employed decision tree ensemble methods that combine random forest (RF), bagging, and random under-sampling boosting to develop a student's academic performance prediction model. The authors in [6] present an empirical comparison of tree-based machine learning algorithms that combine RF, XGBoost, and a C5.0 decision tree to predict the performance of students through modular object-oriented dynamic learning environment (Moodle) data. The authors in [7] utilized several machine learning techniques to estimate student performance. In another study, ref. [8] proposed a support vector machine (SVM)-based model to predict student academic performance. Badal et al. [9] also proposed a machine learning model to predict students' performance. However, many of these approaches are based on conventional machine learning algorithms, and many focus on a single type of educational data, limiting the generalizability of their findings. Moreover, the issue of class imbalance in datasets is often insufficiently addressed, which can significantly impact the performance of predictive models.

To address these challenges, we propose a hybrid deep learning model that combines convolutional neural networks (CNN) and long short-term memory (LSTM) to predict student academic performance. Our model addresses common issues with data preprocessing, such as class imbalance, missing data, and selecting relevant features. Additionally, we apply optimization strategies, including hyperparameter tuning and regularization techniques, to boost predictive accuracy and prevent overfitting. The model is evaluated using two widely benchmarked datasets to ensure performance generalization. The main contributions of this study are summarized below:

- We developed a hybrid deep learning model that combines CNN and LSTM to enhance predictive accuracy in student performance.
- We improved data preprocessing by implementing strategies to address key challenges such as data imbalance and missing values.
- We incorporated advanced optimization and regularization techniques for improved prediction accuracy.
- We compared our model to others, including some of the highest-rated models in the literature.

The rest of the paper is structured as follows: Section 2 presents the related works. Section 3 presents the proposed model, including the experimental setup, the dataset used, data preprocessing methods, and the performance evaluation metrics. Section 4 presents the results and discussions, and Section 5 concludes the paper.

2. Related Works

The recent literature reflects diversity in approaches to student performance prediction. These include tree-based methods, ensemble techniques, shallow machine learning, and deep learning methods.

Pu et al. [10] employed a graph convolutional neural network (GCN) to predict students' academic performance, specifically focusing on students in Chinese-foreign cooperation programs. The aim of the model was to predict students' performance in a course based on their final grades from past semesters. The GCN model captured relationships between students and courses using graph-structured data. The performance of the GCN was compared with the SVM and random forest (RF) models, and it was found that the GCN model yielded an accuracy of 81.5%, which outperformed both the SVM and RF models from the experimentation. Similarly, Poudyal et al. [11] proposed a hybrid 2D CNN model for predicting student performance. The model achieved an accuracy of 88% when compared to traditional machine learning algorithms. In another study, Wang et al. [12] proposed a deep learning approach for predicting student performance, focusing on modeling short-term sequential campus behaviors. The authors combined SVM and recurrent neural networks (RNNs) for the effective capture of behavioral patterns. SVM is used in the first stage to classify static features of student behavior, while RNN is employed in the second stage to model sequential data. The proposed model demonstrated a 45% prediction accuracy. Likewise, Mengash et al. [13] proposed different shallow machine learning techniques that included an artificial neural network (ANN), decision tree (DT), SVM, and NB to improve decision-making in university admissions by predicting student performance. The dataset used was sourced from the computer science department at a Saudi public university. The author achieved the best results from ANN with an accuracy of 79%.

Additionally, Asselman et al. [14] proposed a student performance prediction approach based on analyzing performance factors. The authors employ random forest (RF), AdaBoost, and XGBoost algorithms. The experiment was evaluated on three educational datasets. The authors stated that XGBoost demonstrated the best performance. Turabieh et al. [15] developed an enhanced Harris Hawks Optimization (HHO) algorithm for feature selection in student performance prediction, achieving 92% accuracy. In another study, Yousafzai et al. [16] developed a hybrid neural network approach, combining a bidirectional long short-term memory (BiLSTM) network with an attention mechanism to predict students' future performance based on their past grades. Their model achieved 90.16% accuracy. Mahareek et al. [17] proposed an evolutionary optimized model for enhancing student performance prediction, optimizing the SVM algorithm using simulated annealing, and achieved higher predictive accuracy on three different datasets.

Furthermore, Yağcı et al. [18] used various machine learning algorithms such as SVM, NB, RF, KNN, and LR to predict undergraduate students' final grades in Turkey, achieving 70% accuracy. Keser et al. [19] proposed a hybrid ensemble learning algorithm for predicting student performance in math and Portuguese courses, achieving 96.6% accuracy for math and 91.2% for Portuguese. However, the study did not address challenges like outliers, missing values, and class imbalance, which could affect the model's performance in complex real-world scenarios. Alarape et al. [20] developed a hybrid machine learning model to predict both student academic performance and potential dropout risks. The approach combined SVM and NB algorithms with recursive feature elimination (RFE) to enhance predictive accuracy by selecting the most relevant features. The hybrid model demonstrated improved performance. Table 1 presents a summary of the predictive models using traditional machine learning and deep learning techniques.

Table 1. Comparison of existing methods.

Article	Model	Performance (Acc)
Hai-tao et al. [10]	Graph CNN	81.5%
Poudyal et al. [11]	CNN	88%
Wang et al. [12]	SVM and RNN	86.9%
Mengash et al. [13]	ANN, DT, SVM, NB	79%
Asselman et al. [14]	RF, Adaboost, XGBoost	78.25%, 78.30%, and 78.75%
Turabieh et al. [15]	Harris Hawks Optimization (HHO) Algorithm and Layered RNN	92%
Yousafzai et al. [16]	Attention based-BILSTM	90.16%
Mahareek et al. [17]	SVM	67.77%
Yağcı et al. [18]	SVM, NB, KNN, LR, RF	70%
Keser et al. [19]	GB, XGB, LGB	96.6% and 91.2%
Alarape et al. [20]	SVM and NB	92.73% and 89.09%

The existing studies have employed various traditional machine learning and deep learning algorithms to predict student success. However, traditional machine learning models often treat features independently and lack mechanisms to capture temporal dependencies. Deep learning models like RNNs can analyze sequential information, but they suffer from the vanishing gradient problem, which can lead to the loss of long-term dependencies. CNNs, while effective at extracting local feature patterns, may not fully capture time-dependent trends crucial for modeling student progression. LSTM networks overcome RNN limitations and are better suited to capture temporal dynamics. To bridge this gap, this study proposes a hybrid CNN–LSTM model that combines CNN's ability to extract feature patterns with LSTM's strength in modeling long-range sequential dependencies. This approach aims to better capture the complex relationship features in student performance data. Furthermore, we address issues related to data preprocessing, such as handling data imbalance and addressing missing values, which are often overlooked or insufficiently tackled.

3. Materials and Methods

This section presents the framework of the proposed model, which includes the experimental setup. A brief overview of the datasets employed is provided. In addition, a detailed discussion of the data preprocessing steps, model training, baseline methods, and evaluation metrics used to assess the model performance is presented. The framework used in this study is presented in Figure 1.

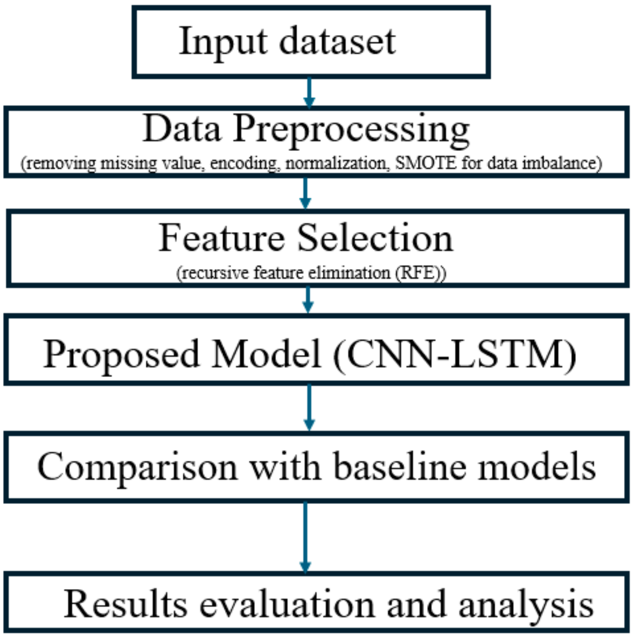
Input dataset
Data Preprocessing
(removing missing value, encoding, normalization, SMOTE for data imbalance)
Feature Selection
(recursive feature elimination (RFE))
Proposed Model (CNN-LSTM)
Comparison with baseline models
Results evaluation and analysis

Figure 1. Workflow of the proposed CNN–LSTM model.

3.1. Description of the Datasets

In this study, we used two publicly available educational datasets.

3.1.1. Open University Learning Analytics Dataset (OULAD)

OULAD is collected from a UK distance-learning institution, which provides detailed data on seven selected courses, students, and students' interactions with the virtual learning environment [21]. The dataset includes learning behavior data for 32,593 students from the 2013–2014 academic year. The dataset tracks seven courses offered across multiple academic disciplines, subdivided into 22 instructional presentations in fields such as science, social sciences, technology, engineering, and mathematics. Each course includes resources in the VLE, logging 20 types of student activities, including content access, discussion forum participation, and assessment attempts [21,22]. Table 2 provides a detailed description of the OULAD dataset.

Table 2. Summary of OULAD dataset.

Module	Domain	Presentations	Students
AAA	Social Sciences	2	748
BBB	Social Sciences	4	7909
CCC	STEM	2	4434
DDD	STEM	4	6272

3.1.2. Western Ontario University (WOU) Dataset

The Western Ontario University (WOU) dataset was obtained from a publicly available GitHub repository. The dataset contains academic performance data from a second-year undergraduate science course at the University. It includes records from 486 students, detailing their scores from various assessments such as assignments, tests, and examinations. The dataset consists of nine academic attributes and a total of 305,933 instances [23].

3.2. Data Preprocessing

Data preprocessing involves several steps prior to training. The first step is checking for missing values. The dataset contains missing features, which are addressed by removing entire records, although several methods have been employed in the literature to address missing values, such as the data imputation approach. However, removing the features enhances the model's confidentiality and trustworthiness, provided the remaining dataset is still huge. The second step involves the use of the sklearn package One-Hot-Encoder to convert the dataset's categorical variables to numeric variables, enabling the data to be used for algorithm training. The third step is data normalization, which is also an important step in data preprocessing. Data normalization is conducted to ensure equal scaling of all features. The min-max scaler in the scikit-learn library was used to normalize the values of the two datasets into the interval [0, 1] set. Mathematically, normalization is expressed as:

$$x\prime = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original value, x' is the normalized value, min (x) is the minimum value in the dataset, and max (x) is the maximum value of the feature. The next step is handling data imbalance. To address this, the synthetic minority oversampling technique (SMOTE) is employed to ensure a balanced training set. It works by picking samples from the minority class, finding their nearest neighbors, and creating new similar samples. The last step is feature selection. The recursive feature elimination (RFE) technique was employed to select relevant features.

3.3. Proposed CNN–LSTM Model

3.3.1. Convolutional Neural Network (CNN)

CNNs are neural networks used for predicting and classifying high-dimensional data using convolution to extract meaningful features from input data. A CNN consists of input, output, and hidden layers, including convolution, activation, and pooling layers [24]. In the convolution layer, a filter or kernel is applied to generate a feature map. This can be mathematically expressed as follows:

$$f(t) = (x \times k)(t) = \sum_{a=-\infty}^{\infty} x(a) \cdot k(t-a) \quad (2)$$

where $x \times k$ represents the convolution operation, $x(a)$ is the input data, and $k(t-a)$ is the kernel function sliding over the input data at each position t. This process enhances computational efficiency by identifying local patterns in the data while conserving memory, particularly when dealing with large datasets. Following the convolution layer, the activation layer applies a nonlinear transformation to the feature maps. In CNN, the rectified linear unit (ReLU) activation function is commonly used, defined as [25]:

$$f(x)\max(0, x) \quad (3)$$

The activation function helps the network to learn nonlinear relationships by passing forward positive features. The pooling layer simplifies the feature maps using max pooling, which selects the maximum value in local regions, enhancing the model's robustness against minor shifts in input data. As training progresses, the network optimizes its weights through backpropagation to minimize the loss function. Once the features are extracted, max pooling is applied again, and the feature maps are flattened into one-dimensional vectors before being passed through a fully connected neural network. Finally, the fully connected layer uses the Softmax activation function, which outputs probabilities for each class. The Softmax function is defined as:

$$p(y = j|z) = \frac{e^{zj}}{\sum_{k=1}^{K} e^{zk}} \quad (4)$$

where $(j|z)$ represents the input to the Softmax for class j, and K denotes the number of classes.

3.3.2. Long Short-Term Memory (LSTM) Network

The LSTM network is a powerful variant of the RNN introduced by S. Hochreiter and J. Schmidhuber in 1997 [26]. Its strength lies in overcoming the limitations of traditional RNNs by addressing the challenge of long-term dependencies, making it useful for tasks like classification and time series prediction [27]. While conventional RNNs tend to struggle with efficiency as the gap length increases, LSTMs maintain performance even with large time gaps [28]. The LSTM network's architecture comprises a memory cell and three key gates: the input, output, and forget gates [28]. The memory cell is responsible for retaining information over extended periods, while the gates regulate the flow of information in and out of the cell. The forget gate is crucial for managing long-term memory, as it determines which information from the previous state should be discarded or retained [29]. The process involves comparing the previous state with the current input and assigning a value between 0 and 1, with 0 indicating information removal and 1 indicating information retention. The input gate determines which new information needs to be stored in the current cell state using activation functions like sigmoid and tanh. The input gate uses the same approach as forget gates. The output gate filters the information from the current cell state, applying an activation function such as tanh to produce the final output [30]. The operation of the gates is mathematically expressed [27] in Equations (5)–(9) as follows:

$$ft = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (5)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \quad (6)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c \tag{7}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \tag{8}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{9}$$

where ft forget gate; σ: sigmoid function; W_f: weight between forget and input gate; h_{t-1}: previous hidden state; x_t: input at the current timestamp; i_t: input gate at time t; W_i , W_c and W_o: weight of the respective gate and tanh operator; tanh: tangent function; o_t: output gate at time t; b_f, b_i, b_o, and b_c: bias; h_t: LSTM output; c_t: cell state.

3.4. Model Tuning

This work combines the strengths of CNN and LSTM to predict student academic performance. The goal is to improve the accuracy of the model and to attain almost real-time prediction. The model is constructed on a Dell Latitude Intel (R) CoreTM i5-4300 processor running at 2.50 GHz and 8 GB of RAM. The model is implemented using the TensorFlow, Pandas, sklearn, and Keras libraries and evaluated using the binary classification methods. To perform the classification, the data are first run through the preprocessing methods and then split into training and testing sets, with 80% for training and 20% for testing. The architecture includes an input layer, a Conv1D layer, LSTM layers, a dense layer, a dropout layer, and an output layer. The CNN handles feature extraction, while the LSTM captures temporal dependencies. The convolutional layer employs the ReLU activation function, with 64 filters and a kernel size of three, while the LSTM layers utilize 100 units. The number of neurons in the output layer matches the number of classes. After the input passes through the convolutional layer, max pooling reduces the feature map to retain essential features. These are then fed into the LSTM layers to extract temporal patterns, followed by a dropout layer to address overfitting. To prevent overfitting, the dropout layer randomly deactivates a fraction of neurons, forcing the remaining ones to adapt and generalize better to unseen data during data training. The ReLU activation function was applied in the convolutional, LSTM, and dense layers, while the Softmax function was used in the output layer. The model's weights were optimized using the Adam optimizer, with categorical cross-entropy as the loss function. We performed a series of experiments, varying the learning rate (0.01, 0.001, 0.0001), and 0.001 was found to be the most effective for maximizing the prediction rate. The model was trained for 10 epochs, and a batch size of 32 yielded the best results after multiple trials. We selected random search based on its efficiency in exploring a larger hyperparameter space with fewer iterations. Table 3 presents the model settings (parameters and hyperparameters) used. The pseudo-code for the CNN–LSTM model is shown in Algorithm 1.

Table 3. CNN–LSTM model parameters and hyperparameters.

Parameters	**Configuration/Value**
Learning rate	0.001
Number of epochs	10
Batch sizes	32
Activation function	ReLU and Softmax
Loss function	Categorical cross-entropy
Optimization algorithm	Adam optimizer
Hyperparameter optimization	Random search
Dropout rate	0.5
Regularization techniques	Dropout technique

3.5. Baseline Methods

To ensure a comprehensive comparison, we evaluated individual classifiers—DNN, CNN, and LSTM—alongside the hybrid CNN–LSTM model. The deep learning algorithms were selected due to their automatic feature extraction capability, sequential dependencies, and computational efficiency. Each was treated as a standalone deep learning approach to highlight the advantages of the hybrid architecture. For the DNN model, we configured three hidden layers with 128, 64, and 32 neurons. The CNN model utilized a 1D-CNN architecture, 64 filters with three kernel sizes, a pooling size of two, and a dense layer containing 100 neurons. For the LSTM model, we set the number of units to 100 with a learning rate of 0.001.

Algorithm 1: CNN–LSTM Model

```
Split data into training and testing sets:
Input: x_train.shape [1]
Output: Metrics
Initialization: Define sequential model: model = Sequential ()
Add Conv1D, (filters, kernel size, activation, MaxPooling1D, LSTM, dropout, Dense)
Model Compile optimizer, learning rate, epochs, batch size
  For epochs = 1 to n do
          while
          train model
          validate model
            monitor = 'loss'
            adjust loss function using categorical cross-entropy
    end while
end for
Evaluate Model
Perform prediction using the model
Calculate metrics
```

3.6. Performance Metrics

Performance metrics, including accuracy, precision, recall, and F-score, are used to assess the effectiveness of proposed models.

3.6.1. Accuracy

Accuracy can be defined as the ratio of correctly predicted instances to the total number of instances. In the context of predicting student performance, it tells us how many students' grades were correctly predicted as either good or weak overall. Accuracy can be defined using Equation (10).

$$\text{Accuracy} = \frac{\text{total correct classified samples}}{\text{total dataset samples}}\% \tag{10}$$

3.6.2. Precision

Precision is the ratio of true positives (correctly predicted good students) to the sum of true positives and false positives (students incorrectly predicted as good). Equation (11) provides the formula for precision.

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{11}$$

3.6.3. Recall (Sensitivity)

Recall or sensitivity measures a model's ability to identify all the relevant students who are actually good. It can be mathematically expressed as Equation (12).

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{12}$$

True positives (TP) represent students correctly predicted as good.

False positives (FP) refer to students incorrectly predicted as good but actually belong to the weak category.

False negatives (FN) predict students who are actually good but were predicted as weak.

3.6.4. F-Score

The F-score is the harmonic mean of precision and recall, providing a balance between the two. It is used to consider both false positives and false negatives. The formula for F-score is presented in Equation (13).

$$Fscore = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{13}$$

4. Results and Discussion

The performance of the hybrid model in predicting student academic outcomes was evaluated using two datasets, OULAD and WOU. Comparisons were made between the results of the hybrid model and those obtained from CNN, LSTM, and DNN models when modelled individually. Additionally, the results were compared with those from existing models in the literature. The training and validation curves for both accuracy and loss are presented in Figure 2. This provides a clearer understanding of how the proposed CNN–LSTM model learns and makes predictions throughout the training process. The curves show that with the increase in the accuracy, the loss score of the model decreases, which shows that as the model's accuracy improves, the errors decrease. Specifically, the decrease in loss values alongside the increase in accuracy confirms that the model is minimizing prediction errors over time while improving its classification performance. Both the training and validation losses show smooth declines from 0.45 and 0.44. This further suggests that the model is not overfitting but rather maintaining a robust generalization capability.

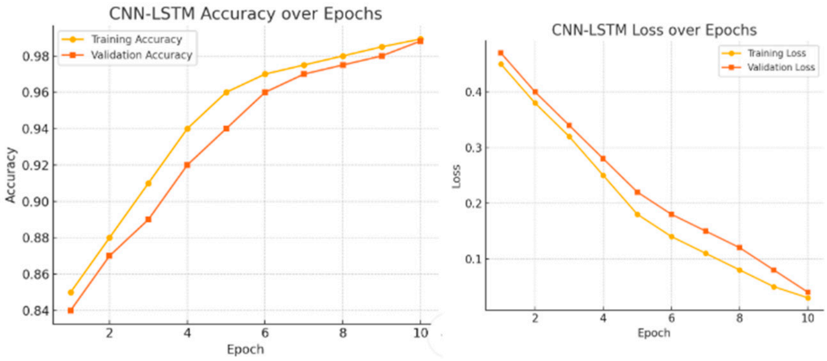

Figure 2. Accuracy and loss analysis of the proposed model.

4.1. Proposed CNN–LSTM Model Performance Using the OULAD Dataset

Using the OULAD dataset, Figure 3 illustrates the performance of the CNN, LSTM, and DNN models when modelled individually, as well as the results from the proposed hybrid model. Out of the single models, the LSTM model performed best, achieving an accuracy of 97.62%, compared to the CNN and DNN, which yielded accuracies of 96.10% and 92.20%, respectively. The LSTM model also achieved a precision of 97.61% and a recall of 97.62%, while the CNN and DNN models had average precision values of 96.13% and 91.70%, and recall values of 96.01% and 92.00%, respectively. The superior performance of the LSTM model can be attributed to its ability to learn complex patterns, especially in structured sequential data.

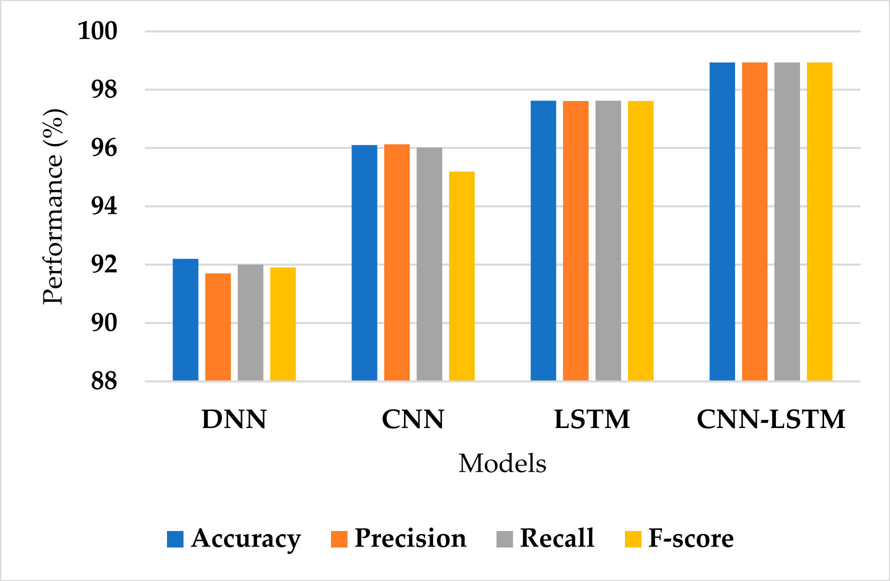

Figure 3. CNN–LSTM results using the OULAD dataset.

Furthermore, as shown in Figure 3 and Table 4, the proposed hybrid model achieved notable results, with an accuracy of 98.93%. This model also presented the best precision, recall, and F-score values. With regards to classification errors, the precision result achieved was 98.93%. The recall result for the developed model was 98.93%. The F-score result, which describes the weighted harmonic mean of the precision and recall results of the developed models, was 98.93%. The hybrid model demonstrated high accuracy, precision, recall, and F-score, with a low number of misclassified instances, confirming its reliability in predicting student performance, reducing both false positives and false negatives.

Table 4. CNN–LSTM results using the OULAD dataset.

Model	Accuracy	Precision	Recall	F-Score
DNN	92.20	91.70	92.00	91.90
CNN	96.10	96.13	96.01	95.19
LSTM	97.62	97.61	97.62	97.61
CNN–LSTM	98.93	98.93	98.93	98.93

4.2. Proposed CNN–LSTM Model's Performance Using the WOU Dataset

Using the WOU dataset, Figure 4 and Table 5 present the classification results for individual models—DNN, CNN, LSTM—and the hybrid model. Compared to the OULAD dataset also employed in this study, the WOU dataset is less voluminous in terms of attributes and instances. Among the individual models, the LSTM demonstrated the highest accuracy of 96%, a precision of 97.45%, a recall of 96%, and an F-score of 92.60%, outperforming the CNN and DNN models, which achieved accuracy levels of 92.20% and 86.52%, precision of 91.80% and 88.84%, recall values of 92% and 86.38%, and F-score values of 91.90% and 87.59%, respectively. In comparison, the CNN and DNN models achieved lower accuracies, respectively. The superior performance of LSTM can be attributed to its well-known advantage in capturing sequential dependencies, which enhances its generalization ability on smaller datasets.

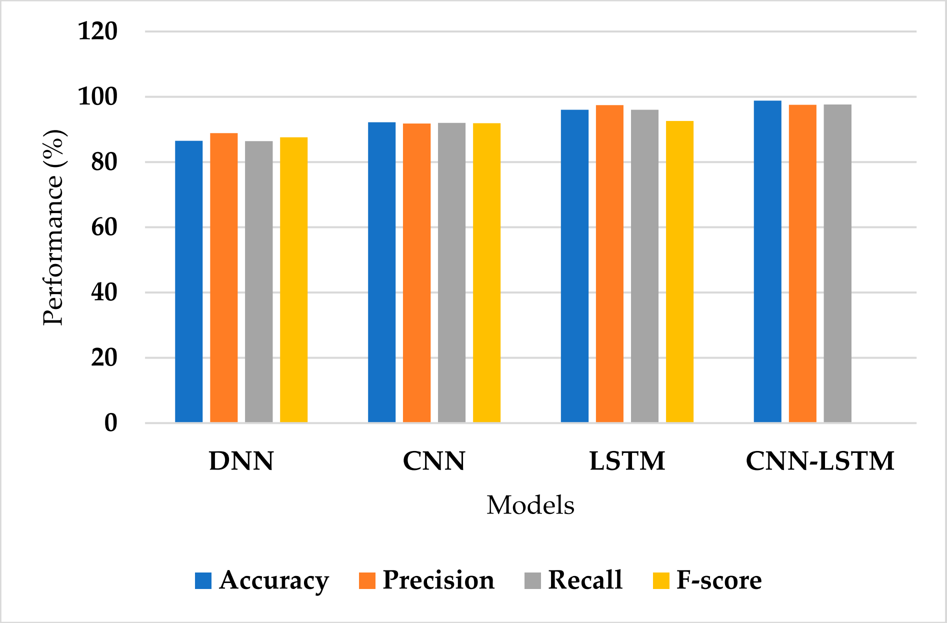

Figure 4. CNN–LSTM results using the WOU dataset.

The hybrid CNN–LSTM model, however, achieved the highest performance, with an accuracy of 98.82%, outperforming all the individual models. With regards to classification error results, the precision result achieved was 97.53%. Regarding the proportion of actual positive instances identified correctly, the developed hybrid model presented a recall result of 97.61%. The F-score result for the developed model was 97.56%. Furthermore, the hybrid model's precision and recall indicate its robustness, resulting in fewer false positives and negatives. This reduction in classification errors emphasizes the advantage of combining the CNN and LSTM algorithms. Overall, these results highlight the hybrid model's potential for superior classification performance, which is driven by effective data preprocessing. Specifically, we addressed class imbalance using the SMOTE technique and managed missing values efficiently. Additionally, selecting relevant training features enhanced the model's efficiency, while the random search optimization approach contributed to the performance of the proposed model.

Table 5. CNN–LSTM binary classification results using the WOU dataset.

Model	Accuracy	Precision	Recall	F-Score
DNN	86.52	88.84	86.38	87.59
CNN	92.20	91.80	92.00	91.90
LSTM	96.00	97.45	96.00	92.60
CNN–LSTM	98.82	97.53	97.61	97.56

4.3. Proposed Model Training and Prediction Time

Table 6 presents the computational efficiency of the proposed CNN–LSTM model compared to the baseline methods, including DNN, CNN, and LSTM. Among the deep learning approaches, the proposed model demonstrated superior efficiency in both training and prediction times. Notably, it achieved prediction in 0.06 s, outperforming the DNN, CNN, and LSTM, which attained 0.09 and 0.08 s prediction, respectively. This efficiency can be attributed to the hybridization of CNN and LSTM, which enhances feature extraction and sequential learning. Importantly, the model maintains a reasonable training time, making it a practical choice for real-time applications.

Table 6. Model training and prediction times.

Model	Training Time (s)	Prediction Time (s)
DNN	0.10	0.09
CNN	0.12	0.08
LSTM	0.14	0.08
CNN–LSTM	0.14	0.06

4.4. Significant Test Results Interpretation

To evaluate the statistical significance of the proposed CNN–LSTM model, we conducted three experiments. First, paired t-tests were used to compare the CNN–LSTM model to each baseline model (DNN, CNN, and LSTM) across performance metrics including accuracy, precision, recall, and F-score. The results indicated that the CNN–LSTM significantly outperforms all baseline models, with *p*-values less than 0.001 in all cases. Secondly, we compared the CNN–LSTM performance across the two datasets (IOULAD and WOU) using a paired *t*-test, which yielded a t-statistic of 3.35 and a *p*-value of 0.044. This indicates a statistically significant difference in the CNN–LSTM performance between datasets at the 5% significance level. Finally, 95% confidence intervals for model accuracy were computed to further support the robustness of our findings, as shown in Table 7.

Table 7. Confidence interval results.

Model	Accuracy	95% Confidence Interval
DNN	92.20	[90.9%, 93.5%]
CNN	96.10	[95.1%, 97.0%]
LSTM	97.62	[96.9%, 98.3%]
CNN–LSTM	98.93	[98.5%, 99.3%]

4.5. Proposed Model Performance Comparison with Results from Similar Studies

Table 8 presents a comparative analysis of the proposed model's results against selected high-performing models from similar studies. Pu et al. [10] employed a graph convolutional neural network (GCN), achieving an accuracy of 81.5%. In another study, Poudyal et al. [11] developed a hybrid 2D convolutional neural network (CNN) model, which attained an accuracy of 88%. Mengash et al. [13] evaluated various machine learning algorithms, including artificial neural networks (ANN), decision trees (DT), support vector machines (SVM), and naïve Bayes (NB), with ANN yielding the highest accuracy at 79%. Wang et al. [12] combined SVM and recurrent neural networks (RNN) to model student behavioral patterns, resulting in a prediction accuracy of 45%. Yağcı et al. [18] achieved an accuracy of 70% using SVM, NB, random forest (RF), k-nearest neighbors (KNN), and logistic regression (LR). While utilizing ensemble methods such as gradient boosting, extreme gradient boosting, and light gradient boosting machine (LightGBM), the authors in [19] achieved a commendable predictive accuracy of 96.6% in math and 91.2% overall. However, their study did not adequately address challenges related to missing values and class imbalance, which limits the robustness of their findings. In a similar hybrid model, Yousafzai et al. [16] utilized a combination of bidirectional long short-term memory (BiLSTM) and an attention mechanism to predict student performance, achieving an accuracy of 90%. Although these models demonstrate generally satisfactory results, few have evaluated their performance across multiple datasets, which is crucial for assessing generalizability. In contrast, the proposed hybrid model not only outperformed these highly rated models but also proved its effectiveness in both accuracy and computational efficiency.

Table 8. Performance Comparison with Previous Studies.

Articles	Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
[10]	GCNN	81.5	-	-	-
[13]	ANN	79.22	81.44	78.03	79.70
[16]	BiLSTM-AM	90.16	90	90	90
[19]	GB, XGBoost, and LightGBM	92.40, 94.13, 89.07	-	-	92.32, 94.00, 88.91
[11]	CNN	88	-	-	-
[12]	SVM-RNN	86.90	-	81.57	-
[18]	NN and RF	74.6	74.8	74.6	72.3
Our Work	DNN	92.20	91.70	92.00	91.90
	CNN	96.10	96.13	96.01	95.19
	LSTM	97.62	97.61	97.62	97.61
	CNN–LSTM	98.93	98.93	98.93	98.93

5. Conclusions

In this study, we develop a robust hybrid deep learning model—CNN–LSTM—to accurately predict student academic performance while addressing key data-preprocessing limitations common in educational datasets. The proposed model was trained and validated on a preprocessed dataset and evaluated using standard metrics. The experimental results revealed that the proposed CNN–LSTM model outperformed existing methods across all evaluation metrics. Notably, it achieved an accuracy of 98.93%, with consistently high precision, recall, and F-scores, highlighting its effectiveness in capturing academic performance patterns. These findings emphasize the potential of deep learning techniques, particularly hybrid models, in enhancing the accuracy and efficiency of student success prediction. Future research will explore the integration of attention mechanisms with advanced deep learning architectures to further enhance prediction accuracy. We plan to explore methods such as SHAP (Shapley additive explanations) to provide deeper insights into the model's decision-making process. Additionally, evaluating model generalizability across diverse student datasets will be a focus.

Author Contributions: Conceptualization, K.O.A. and M.B.M.; methodology, K.O.A.; software, K.O.A.; validation, K.O.A. and M.B.M.; formal analysis, K.O.A. and M.B.M.; investigation, K.O.A.; resources, M.B.M.; data curation, K.O.A.; writing—original draft preparation, K.O.A.; writing—review and editing, K.O.A. and M.B.M.; visualization, K.O.A. and M.B.M.; supervision, M.B.M.; project administration, M.B.M.; funding acquisition, M.B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Directorate at Mangosuthu University of Technology. The APC was funded by the Research Directorate at Mangosuthu University of Technology.

Data Availability Statement: Publicly open-source datasets were analyzed in this study. This data can be accessed at: https://www.kaggle.com/datasets/anlgrbz/student-demographics-online-education-dataoulad (accessed on 10 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
EDM	Educational data mining
DL	Deep learning
DNN	Deep neural network
LSTM	Long short-term memory
OULAD	Open University Learning Analytics Dataset
WOU	Western Ontario University

References

1. Pelima, L.R.; Sukmana, Y.; Rosmansyah, Y. Predicting university student graduation using academic performance and machine learning: A systematic literature review. *IEEE Access* **2024**, *12*, 23451–23465. [CrossRef]
2. Angeioplastis, A.; Aliprantis, J.; Konstantakis, M.; Tsimpiris, A. Predicting Student Performance and Enhancing Learning Outcomes: A Data-Driven Approach Using Educational Data Mining Techniques. *Computers* **2025**, *14*, 83. [CrossRef]
3. Ashish, L.; Anitha, G. Transforming Education: The Impact of ICT and Data Mining on Student Outcomes. In Proceedings of the 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 8–9 August 2024; Volume 1, pp. 675–683.
4. Malik, S.; Patro, S.G.K.; Mahanty, C.; Hegde, R.; Naveed, Q.N.; Lasisi, A.; Buradi, A.; Emma, A.F.; Kraiem, N. Advancing Educational Data Mining for Enhanced Student Performance Prediction: A Fusion of Feature Selection Algorithms and Classification Techniques with Dynamic Feature Ensemble Evolution. *Sci. Rep.* **2025**, *15*, 8738. [CrossRef] [PubMed]
5. Ahmad, A.; Ray, S.; Khan, M.T.; Nawaz, A. Student Performance Prediction with Decision Tree Ensembles and Feature Selection Techniques. *J. Inf. Knowl. Manag.* **2025**, *24*, 2550016. [CrossRef]
6. Rogers, J.K.; Mercado, T.C.; Cheng, R. Predicting Student Performance Using Moodle Data and Machine Learning with Feature Importance. *Indones. J. Electr. Eng. Comput. Sci.* **2025**, *37*, 223–231. [CrossRef]
7. Mohammad, A.S.; Al-Kaltakchi, M.T.; Alshehabi Al-Ani, J.; Chambers, J.A. Comprehensive evaluations of student performance estimation via machine learning. *Mathematics* **2023**, *11*, 3153. [CrossRef]
8. Bisri, A.; Heryatun, Y.; Navira, A. Educational Data Mining Model Using Support Vector Machine for Student Academic Performance Evaluation. *J. Educ. Learn. (EduLearn)* **2025**, *19*, 478–486. [CrossRef]
9. Badal, Y.T.; Sungkur, R.K. Predictive modelling and analytics of students' grades using machine learning algorithms. *Educ. Inf. Technol.* **2023**, *28*, 3027–3057. [CrossRef]
10. Pu, H.; Fan, M.; Zhang, H.; You, B.; Lin, J.; Liu, C.; Zhao, Y.; Song, R. Predicting academic performance of students in Chinese-foreign cooperation in running schools with graph convolutional network. *Neural Comput. Appl.* **2021**, *33*, 637–645.
11. Poudyal, S.; Mohammadi-Aragh, M.J.; Ball, J.E. Prediction of student academic performance using a hybrid 2D CNN model. *Electronics* **2022**, *11*, 1005. [CrossRef]
12. Wang, X.; Yu, X.; Guo, L.; Liu, F.; Xu, L. Student performance prediction with short-term sequential campus behaviors. *Information* **2020**, *11*, 201. [CrossRef]
13. Mengash, H.A. Using data mining techniques to predict student performance to support decision making in university admission systems. *IEEE Access* **2020**, *8*, 55462–55470. [CrossRef]
14. Asselman, A.; Khaldi, M.; Aammou, S. Enhancing the prediction of student performance based on the machine learning xgboost algorithm. *Interact. Learn. Environ.* **2023**, *31*, 3360–3379. [CrossRef]
15. Turabieh, H.; Azwari, S.A.; Rokaya, M.; Alosaimi, W.; Alharbi, A.; Alhakami, W.; Alnfiai, M. Enhanced harris hawks optimization as a feature selection for the prediction of student performance. *Computing* **2021**, *103*, 1417–1438. [CrossRef]
16. Yousafzai, B.K.; Khan, S.A.; Rahman, T.; Khan, I.; Ullah, I.; Ur Rehman, A.; Baz, M.; Hamam, H.; Cheikhrouhou, O. Student-performulator: Student academic performance using hybrid deep neural network. *Sustainability* **2021**, *13*, 9775. [CrossRef]
17. Mahareek, E.A.; Desuky, A.S.; El-Zhni, H.A. Simulated annealing for svm parameters optimization in student's performance prediction. *Bull. Electr. Eng. Inform.* **2021**, *10*, 1211–1219. [CrossRef]

18. Yağcı, M. Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learn. Environ.* **2022**, *9*, 11. [CrossRef]
19. Keser, S.B.; Aghalarova, S. Hela: A novel hybrid ensemble learning algorithm for predicting academic performance of students. *Educ. Inf. Technol.* **2022**, *27*, 4521–4552. [CrossRef]
20. Alarape, M.A.; Ameen, A.O.; Adewole, K.S. Hybrid students' academic performance and dropout prediction models using recursive feature elimination technique. In *Advances on Smart and Soft Computing*; Springer: Singapore, 2022; pp. 93–106.
21. Kuzilek, J.; Hlosta, M.; Zdrahal, Z. Open University learning analytics dataset. *Sci. Data* **2017**, *4*, 170171. [CrossRef]
22. Jin, L.; Wang, Y.; Song, H.; So, H.J. Predictive Modelling with the Open University Learning Analytics Dataset (OULAD): A Systematic Literature Review. In Proceedings of the International Conference on Artificial Intelligence in Education, Recife, Brazil, 8–12 July 2024; Springer Nature: Cham, Switzerland, 2024; pp. 477–484.
23. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Systematic ensemble model selection approach for educational data mining. *Knowl.-Based Syst.* **2020**, *200*, 105992. [CrossRef]
24. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
25. Banerjee, C.; Mukherjee, T.; Pasiliao, E. Feature representations using the reflected rectified linear unit (RReLU) activation. *Big Data Min. Anal.* **2020**, *3*, 102–120. [CrossRef]
26. Mienye, I.D.; Swart, T.G.; Obaido, G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information* **2024**, *15*, 517. [CrossRef]
27. Adefemi Alimi, K.O.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, O.A. Refined LSTM Based Intrusion Detection for Denial-of-Service Attack in Internet of Things. *J. Sens. Actuator Netw.* **2022**, *11*, 32. [CrossRef]
28. Shiri, F.M.; Perumal, T.; Mustapha, N.; Mohamed, R. A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv* **2023**, arXiv:2305.17473.
29. Fang, W.; Chen, Y.; Xue, Q. Survey on research of RNN-based spatio-temporal sequence prediction algorithms. *J. Big Data* **2021**, *3*, 97. [CrossRef]
30. Gaur, D.; Kumar Dubey, S. Development of activity recognition model using lstm-rnn deep learning algorithm. *J. Inf. Organ. Sci.* **2022**, *46*, 277–291.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.