Journal of Computer Science

Original Research Paper

Deep Learning Perspective on Assessing and Elevating Engineering Student's Performance

[1]Kandula Neha and [2]Ram Kumar

[1]Department of Computer Science and Engineering Lovely Professional University, Punjab, India
[2]Department of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh, Bhopal, India

Article history
Received: 14-02-2024
Revised: 15-06-2024
Accepted: 29-06-2024

Corresponding Author:
Ram Kumar
Department of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh, Bhopal, India
Email: drramkumar.research@gmail.com

Abstract: In addressing the need for successful frameworks to break down understudy execution, this study presents a profound learning-based approach for thorough understudy execution examination inside instructive establishments. The framework intends to evaluate understudies' presentation levels and distinguish those qualified for positions, needing extra help, or in danger of exiting. Utilizing a Long Momentary Memory (LSTM) model, a sort of intermittent brain organization (RNN), the proposed framework predicts fourth-year understudies' presentation by utilizing three years of verifiable understudy marks information to catch fleeting examples and conditions. Broad testing and assessment show the LSTM model's surprising exactness, accomplishing an accuracy of 99.8% in distinctive understudies' exhibition levels. Through the force of profound realizing, this framework engages instructive establishments to precisely separate between high-performing, low-performing, and in-danger understudies, working with vocation arranging and giving designated open doors to understudy positions. In addition, it promotes good help and mediation for students who are at risk of dropping out and improving real standards. By introducing deep learning strategies, especially LSTM models, this research provides valuable experience and direct prospects for investigating the implementation of non-earning people, empowering learning organizations to follow making informed choices, and showing direction and mediation. Finally, the framework that is being developed can improve the result of education without achieving it by enhancing dynamic changes and encouraging individual contributions in educational areas.

Keywords: Student Performance Analysis, Deep Learning, LSTM Model, Good Performers, Poor Performers, Student Support, Placement Eligibility, Dropout Prediction

Introduction

In today's instructive environment, there is a rising center on utilizing data-driven activities to make strides in instructive results and understudy execution. Viable analysis of understudy information to discover designs, patterns, and components that impact a student's success or failure could be an issue for instructive education. This think about offers a profound learning-based understudy execution investigation framework that particularly centers on utilizing an LSTM demonstrate for execution expectation in arrange to address this issue.

In arranging for instructive teachers to offer individualized help and mediation, it is fundamental to be able to estimate understudy execution dependably. Educators can take uncommon activities to meet the necessities of understudies who require additional offer assistance by recognizing them and executing centered activities, which inevitably upgrade understudy comes about. Comparable to that, recognizing understudies who are accomplishing scholarly victory or who qualify for arrangement chances empowers schools to offer vital exhortation and create their potential assistance.

Due to its capacity to precisely speak to complicated designs and relationships in information, profound learning, a department of machine learning, has pulled in a parcel of intrigue as of late. Successive information-preparing assignments have been effectively completed by one prevalent profound learning demonstration, the LSTM. LSTMs can capture long-term conditions, which makes them suitable for surveying understudy execution information with worldly designs.

The objective of this study is to form a careful framework for analyzing student performance that creates utilization of profound learning, more particularly the LSTM demonstration, to estimate how fourth-year understudies would do. The strategy looks to donate exact estimates of understudy execution, separating between high achievers, moo achievers, and understudies in require of assist offer assistance by taking under consideration the marks information from the past three a long time. The innovation moreover identifies children who are qualified for arrangement conceivable outcomes and those who are at peril of taking off school, permitting proactive intercessions and counseling.

By utilizing the qualities of profound learning and LSTM models, this investigation seeks to essentially progress the consideration of understudy execution investigation by giving instructive teach an effective device for comprehending, determining, and advancing understudy accomplishment. By empowering evidence-based decision-making for way better instructive results, the made framework has the potential to alter how educate approach understudy execution examination.

Related Work

The assessment method in engineering education comprises semester-end/external exams and ongoing internal evaluation. However, relying solely on the pass rate may not provide a comprehensive understanding. To address this issue and identify the major variables affecting engineering students' academic performance, (Venkatesh, 2013) has developed a recommended analytical technique. This technique involves creating a mathematical expression to assist with analysis, enabling self-evaluation, identifying areas for improvement, and implementing corrective actions. The objective is to enhance teaching and learning methods, promote growth in technical institutions and engineering colleges, and improve overall academic performance.

Arsad *et al*. (2023) study focuses on the importance of predicting academic performance in engineering courses to enable strategic intervention before students reach higher semesters, including the final semester before graduation. The study utilizes the Cumulative Grade Point Average (CGPA) as an indicator of academic achievement in the eighth semester. Two models, Artificial Neural Network (ANN) and Linear Regression (LR), are employed to predict academic performance. The study uses the foundational subjects from the first semester as independent variables and assesses model performance using the coefficient of correlation (R) and mean square error (MSE). The findings reveal a significant association between the fundamental outcomes for core courses in semesters one or three and the final CGPA. These findings have implications for educational organizations aiming to enhance student performance, instructional methods, and learning environments.

In recent times, poor academic performance has become a widespread issue faced by many engineering colleges. Due to the continuous decline in pass rates, it is crucial to identify the variables influencing academic performance and develop forecasting models. The study (Bithari *et al*., 2020) aims to forecast the academic success of engineering students based on their prior academic records, demographic data, familial histories, and other relevant characteristics. The study creates a predictive model using conventional classifiers like decision tree, SVM, and linear regression. Additionally, the ensemble technique known as voting is employed to enhance the performance of individual classifiers. The study reveals that using the ensemble voting approach leads to significantly higher accuracy, precision, recall, and F1-score. Data for the study were collected directly from the personal files of pass-out students of Paschim Anchal Engineering Campus, Pokhara, between the years 2004-2015.

In their study, (Mothe *et al*., 2019) discussed various research methodologies employed for evaluating student performance using educational analytics tools. The authors highlighted the importance of leveraging data analytics techniques to gain insights into student performance and make informed decisions in educational settings.

Firdausiah Mansur *et al*. (2019) in their study focused on the application of deep learning algorithms to analyze student behavior and develop personalized learning models. The authors emphasized the role of deep learning in understanding student characteristics and tailoring educational interventions to enhance their learning outcomes. (Waheed *et al*., 2020) explored the use of deep learning models for predicting student academic performance based on Virtual Learning Environment (VLE) big data. The potential of deep learning in leveraging large-scale educational data to accurately predict student outcomes and provide early intervention strategies was highlighted in their research.

In their study, Hussein Altabrawee *et al*. (2019) discuss the application of various machine-learning techniques to predict student performance. They compare the performance of different models and emphasize the importance of accurate prediction for effective educational planning and intervention. Almayan and Mayyan (2016) proposed a model to improve the accuracy of predicting students' final grades using the Particle Swarm Optimization (PSO) algorithm. Their study mainly focused on enhancing the prediction model through optimization techniques, providing valuable insights into improving the accuracy of student performance prediction.

Sultana *et al*. (2019), in their study titled, explored the application of deep learning and data mining methods for predicting student performance. The authors highlighted the benefits of utilizing these techniques to extract meaningful patterns from student data and make accurate predictions about their academic performance. Yanamandra and Prasad (2022) presented a feasibility study for student performance prediction using machine learning in their study. The authors explored the application of machine learning techniques in predicting student academic performance. The study assesses the feasibility of using machine-learning algorithms to accurately predict student performance based on various factors.

In their research study, Neha and Sidiq (2020) explored the analysis of student academic performance using expert systems. They investigated the application of artificial intelligence techniques to predict and understand student performance. By leveraging expert systems, the authors aimed to provide valuable insights into factors that influence student success. The study highlights the potential of expert systems in educational settings and emphasizes their role in enhancing student performance analysis.

Kaur and Kaur (2023) proposed a prediction model for student academic performance using machine learning-based analytics. Their study emphasizes the use of machine learning techniques in developing a model that can predict student performance. They highlighted the importance of leveraging machine-learning algorithms to extract meaningful insights from educational data and improve predictions regarding student academic performance.

In their systematic literature review, Albreiki *et al*. (2021) explored the use of machine learning techniques for predicting student performance. The authors analyze existing research in this area to gain insights into the effectiveness and applicability of machine learning algorithms in predicting student outcomes. The review provides a comprehensive overview of the current state of the field and identifies trends, challenges, and potential future directions for research in student performance prediction.

Neha's (2021) study focuses on the prediction of student academic performance based on expert systems. The study highlights the significance of expert systems in accurately predicting student outcomes. By utilizing data-driven approaches and expert knowledge, the author develops a predictive model to identify patterns and factors that influence student performance. The research contributes to the field of educational data analysis by providing insights into the effectiveness of expert systems in predicting and improving student academic performance.

Jabbar *et al*. (2022) focused on student performance prediction in e-learning environments using machine learning. Their study investigates the application of machine learning algorithms in predicting student performance specifically in the context of online learning. The authors explore the potential of these algorithms to analyze student data, identify patterns, and make accurate predictions about student outcomes, thereby assisting educators in providing personalized support and interventions.

In their research study, Neha *et al*. (2023) propose a deep neural network model for identifying predictive variables and evaluating student academic performance. The authors aim to leverage the power of deep learning techniques to uncover hidden patterns and relationships within student data. By employing advanced machine learning algorithms, the study demonstrates the effectiveness of the proposed model in predicting and evaluating student academic performance. The research offers valuable insights into the application of deep neural networks in educational contexts, contributing to the development of data-driven approaches for student performance analysis (Neha *et al*., 2021).

The above review of the literature highlights the increasing interest in utilizing data-driven approaches to analyze student data, extract meaningful patterns, and make accurate predictions about academic performance. The studies reviewed in this section demonstrate the effectiveness of ML algorithms, such as artificial neural networks, linear regression, and support vector machines, in predicting student performance. Additionally, the application of deep learning techniques, such as convolutional neural networks and recurrent neural networks, shows promising results in capturing complex patterns and temporal dependencies in student data. Recently researchers have explored the use of expert systems, which combine domain knowledge and rule-based reasoning, for predicting student performance (Neha and Kumar, 2023). These systems offer a structured and interpretable approach to analyzing student data, providing valuable insights for educational institutions to support and intervene with at-risk students.

Deep learning is chosen as the base of this research among the various approaches for predicting student performance due to its ability to capture complex patterns and temporal dependencies in student data, offering a higher potential for accurate predictions. From the various deep learning techniques, Long Short-Term Memory (LSTM) is chosen as a model for predicting student performance due to its unique architecture designed to capture long-term dependencies in sequential data. LSTM networks are well suited for analyzing temporal relationships and patterns in student academic data, making them effective in predicting future performance based on historical information.

Materials and Methods

Engineering (bachelor of technology/ bachelor of engineering) students in India undergo an 8-semester system (4 years) and have an evaluation at the end of each semester. The final year of engineering is very crucial in a student's life and for the educational institution as it is in their last two semester, they undergo placements. Placements are crucial for a student’s success in career and it is considered a matter of reputation for colleges.

The student dropout rate is also high in these semesters. Hence, we aim to find out good performers, poor performers, Students who require academic guidance, students who are directly eligible for placements, and students who are prone to drop out, at the end of 3rd year or 6th semester based on cumulative percentages from all the 6 semesters. This can aid in handling the student dropouts well in advance, devising timely plans for supporting the students in their academics, and guiding them well through their placements.

The algorithms and approaches used in this study to create the deep learning-based student performance analysis system, specifically the LSTM model, are detailed in this section. Data preprocessing, LSTM model architecture, and evaluation metrics are discussed in Fig. (1).

Data Preprocessing

The first step in the methodology involved data preprocessing to ensure the quality and suitability of the dataset for deep learning analysis. The following steps were performed.

Data Cleaning

The dataset, obtained from a Deemed University, underwent thorough cleaning to handle missing values and outliers. Missing values were addressed through appropriate techniques such as imputation or exclusion, while outliers were detected and treated using suitable statistical methods.

Feature Selection

To focus on relevant information, feature selection techniques were employed to identify the most important features for predicting student performance. This step helped reduce the dimensionality of the dataset and improve model efficiency.

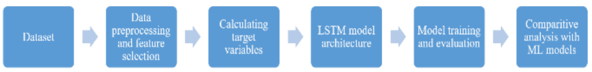

Fig. 1: Flow of study

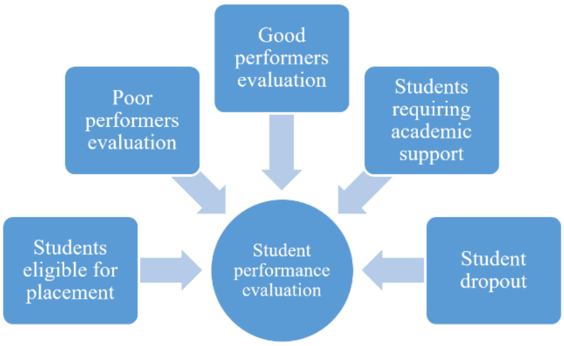

Fig. 2: Target variables

Data Normalization

To facilitate fair treatment of different features and enhance model convergence during training, data normalization techniques were applied. Normalization methods, such as min-max scaling or z-score normalization, were employed to bring the numerical attributes within a standardized range.

Evaluating Target Variables

We aim to predict the target variables mentioned below in order to evaluate the student's performance in the coming final year of academics. For this purpose, we consider the 3 years (6 semesters percentages) (Fig. 2):

- Good performers: Individuals who consistently achieve high standards or exceed expectations in academic or professional settings
- Poor performers: Individuals who consistently demonstrate below-average results or fail to meet expected standards in academic or professional endeavors
- Support required in academics: The need for assistance, guidance, or additional resources to improve academic performance or overcome challenges
- Eligibility for placement: Meeting the criteria or requirements necessary to participate in job placements or recruitment processes
- Student dropout: The act of a student discontinuing or leaving a program of study or educational institution before completion

Algorithm 1: An algorithm for evaluating target variables

Input: 6 Semester percentages:
k = semester_grades = [a, b, c, d, e, f]
h = extracurricular activities
i = Academic awards and achievements
j = Coding skills
Output: Evaluating and assigning Target variables
Step-1: Evaluate Good Performers
good_performers = 1 if all (grade >60 for grade in k)
else 0
Step 2: Evaluate Poor Performers
poor_performance = 1 if max (k) <40
else 0 Step-3: Evaluate Students who require support
support required = 1 if any (40> grade <60 for grade in k)
else 0
Step 4: Evaluate Students eligible for placements
eligible_for_placement = 1 if all (grade >65 for grade in k) and (j or i or h)
else 0
Step-5: Evaluate Student Dropout
dropout = 1 if min(k) <35 and g <30
else 0

The algorithm provides a systematic approach for evaluating student performance and assigning target variables based on semester grades and additional factors such as extracurricular activities, academic achievements, and coding skills. By categorizing students into different groups, educational institutions can identify those in need of support, eligible for placements, or at risk of dropout, enabling targeted interventions and support strategies to improve student outcomes.

LSTM Model Architecture

The Long Short-Term Memory (LSTM) model was chosen for its ability to capture long-term dependencies in sequential data. LSTM is a type of Recurrent Neural Network (RNN) architecture that is particularly effective in capturing long-term dependencies in sequential data. It overcomes the vanishing gradient problem that can occur in traditional RNNs, allowing it to remember and utilize information from earlier time steps in the sequence.

This LSTM model consists of three layers Input, hidden, and output layers showed in Fig. (3).

1. Input layer: This is the first layer of the LSTM network, which receives the input data. The number of units in this layer is determined by the dimensionality of the input data
2. The model includes multiple LSTM layers stacked to enhance its ability to learn complex patterns from the input sequences

- Layer 1 (LSTM)
  - Units: 128 units
  - Dropout rate: 0.2
  - Recurrent dropout rate: 0.2
  - Activation function: Tanh
  - Recurrent activation function: Sigmoid
  - Input shape: (None, D)
- Layer 2 (LSTM)
  - Units: 64 units
  - Dropout rate: 0.2
  - Recurrent dropout rate: 0.2
  - Activation function: Tanh
  - Recurrent activation function: Sigmoid
  - Return sequences: True (to allow stacking of additional LSTM layers)
- Layer 3 (LSTM)
  - Units: 32 units
  - Dropout rate: 0.2
  - Recurrent dropout rate: 0.2
  - Activation function: Tanh
  - Recurrent activation function: Sigmoid
  - Return sequences: False (output only the last output in the output sequence)

3. Fully connected layer

- Dense layer
  - Units: 1 (or the number of output classes for classification tasks)
  - Activation function: Linear (for regression tasks) or softmax/sigmoid (for classification tasks)

4. Output layer: The output layer of the LSTM network produces the final predictions or classifications based on the information processed by the LSTM layers. The number of units in this layer depends on the specific task (e.g., regression, classification) and the desired output dimensionality

Generalized formulas in an LSTM model:

1. Input gate (i) formula:

$$i[t] = (Wix[t] + Uih[t-1] + bi)$$

- Explanation: The input gate controls the flow of information into the cell state ($C[t]$). It determines which parts of the input and previous hidden state are relevant for the current time step

2. Forget gate (f) formula:

$$f[t] = \sigma Wfx[t] + Ufh[t-1] + bf)$$

- Explanation: The forget gate determines which information to discard from the cell state. It decides how much of the previous cell state ($C[t\text{-}1]$) to retain for the current time step

3. Candidate cell state ($\hat{C}[t]$) formula:

$$\hat{C}[t] = \tanh(Wcx[t] + Uch[t-1] + bc)$$

- Explanation: The candidate cell state calculates a new candidate value that can be added to the cell state. It combines the current input and previous hidden state

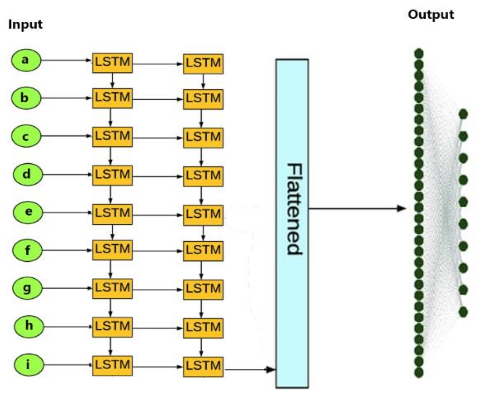

Fig. 3: LSTM model architecture of the proposed system

4. Cell state (*C[t]*) formula:

$$C[t] = f[t] * C[t-1] + i[t] * \hat{C}[t]$$

- Explanation: The cell state represents the memory of the LSTM. It is updated based on the forget gate and the input gate, incorporating relevant information from the previous cell state and the candidate cell state

5. Output gate (o) formula:

$$o[t] = \sigma(Wox[t] + Uoh[t-1] + bo)$$

- Explanation: The output gate determines which parts of the cell state should be output as the hidden state ($h[t]$) of the LSTM at the current time step

6. Hidden state ($h[t]$) formula:

$$h[t] = o[t] * \tanh(C[t])$$

- Explanation: The hidden state represents the output of the LSTM at the current time step. It is calculated by applying the output gate to the cell state

In an LSTM model, the above formulas are applied iteratively for each time step in the sequence, allowing the model to capture long-term dependencies and make predictions based on the sequential input data.

LSTM models are widely used in various applications, including natural language processing, speech recognition, time series analysis, and, as in this case, student performance analysis. Their ability to handle long-term dependencies and effectively model sequential data makes them suitable for tasks that involve analyzing and predicting patterns in sequential data.

Algorithm 2: Proposed LSTM for evaluating student academic performance

LSTM Input Features: x = [a, b, c, d, e, f, g, h, i, j]
Output: A binary classification indicating the assessment of Target variables
Step 1: Build and Compile LSTM
Compile the LSTM model with the following parameters:
- Optimizer: Adam
- Loss function: Binary cross entropy
- Metrics: Accuracy, precision, recall, F1-score

```
model. compile (optimizer, loss,
metrics = ['accuracy', metrics.precision(), metrics.recall(),
metrics. F1-score ()])
```

Step 2: Train LSTM

```
model.fit(X_train, y_train, epochs, batch_size)
```

Step 3: Evaluate the LSTM model and print accuracies of trained LSTM model.

```
accuracy, precision, recall, f1 = model.evaluate (X_test, y_test)
```

Evaluation Metrics

To assess the performance of the LSTM model, several evaluation metrics were employed.

Accuracy

Accuracy measures the proportion of correctly predicted student performance categories. It provides an overall assessment of model performance.

The formula for accuracy using *True Positives* (*TP*), *False Positives* (*FP*), and *True Negatives* (*TN*):

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Precision, Recall, and F1-Score

Precision, recall (also known as sensitivity or true positive rate), and F1-score were calculated to evaluate the model's performance on each performance category individually. These metrics provide insights into the model's ability to correctly identify specific categories while considering both false positives and false negatives:

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

$$F1 - Score = 2 * (\mathrm{Pr}ecision * \mathrm{Re}call)/(\mathrm{Pr}ecision + \mathrm{Re}call)$$

Training and Validation

The preprocessed dataset was split into training and validation sets. The training set was used to train the LSTM model, while the validation set was utilized to monitor the model's performance and prevent overfitting.

Model Training

The LSTM model was trained using the Adam optimization algorithm and the binary cross-entropy loss function. The model underwent iterative training epochs to minimize the loss and update the weights.

Model Evaluation

After training, the LSTM model's performance was evaluated Various evaluation metrics were used to assess our LSTM models' performance. These metrics shed light on how well the models did at solving the classification problem. The key assessment measurements utilized in this examination incorporate accuracy, Precision, recall, and F1-score. We can make well-informed comparisons and acquire a comprehensive knowledge of the performance of each model by looking at these metrics.

Model Fine-Tuning and Iteration

The LSTM model's performance was improved by fine-tuning and iterating on the basis of the evaluation results. This required adjusting hyperparameters like the dropout rate, batch size, learning rate, and the number of

DOI: 10.3844/jcssp.2024.1455.1469

LSTM layers. Various cycles were led, each time surveying the model's presentation on the approval set and making fitting changes.

Implementation

The total number of good performers in any given academic setting is an important metric for evaluating the overall success of the institution and the quality of education being provided. A barplot can be an effective way to visualize and understand this metric.

The formula can be written as:

$$dropout = 1 if \min(k) < 35 \; and \; g < 30 \; else \; 0$$

where:

- "k" represents a list of grades
- "*Min*(k)" returns the lowest value in the list "k"
- "*g*" represents an individual grade for a specific subject
- "*Dropout*" is a binary variable that is assigned a value of 1 if the lowest grade in "k" is less than 35 and the grade for the specific subject is less than 30 and 0 otherwise

The formula checks if the lowest grade in "k" is less than 35, using the "min" function, and if the grade for the specific subject is less than 30. If both conditions are met, the formula assigns a value of 1 to the variable "*dropout*", indicating that the student is at risk of dropping out. Otherwise, it assigns a value of 0, indicating that the student is not at risk of dropping out.

In a plot of the total number of good performers, the X-axis represents the branch and the Y-axis represents the number of students who are classified as good performers based on a set of predefined criteria such as high grades, good attendance, and extracurricular activities:

$$good - performance = 1 \; if \; all(\; grades > 60 \; for \; a \; grade \; in \; k) \; else \; 0$$

where:

- "k" represents a list of grades
- "*Grade*" represents an individual grade in the list "k"
- "All (grade >60 for a grade in k)" checks if all grades in the list "k" are greater than 60
- "*Good performance*" is a binary variable that is assigned a value of 1 if all grades in "k" are greater than 60 and 0 otherwise

The formula uses a list comprehension to iterate over each grade in the list "k" and check if it is greater than 60. The "*all*" function returns True if all of the conditions are met and False otherwise. If all grades are greater than 60, the formula assigns a value of 1 to the variable "*good performance*", otherwise it assigns a value of 0. To determine the total number of poor performers among the students, the same algorithm as for the total number of excellent performers can be used, with minor modifications to the criteria for identifying poor performers. In this case, a pupil is deemed a weak performer if their cumulative semester grade point average is below 40%.

Using the same dataset of student academic performance metrics, the algorithm can be applied to determine the total number of weak performers. The resulting scatter plot will depict the distribution of low-achieving students among all students.

The formula can be written as:

$$poor - performance = 1 \; if \; \max(k) < 40 \; else \; 0$$

where:

- "k" represents a list of grades
- "*Max*(k)" returns the highest value in the list "k"
- "*Poor performance*" is a binary variable that is assigned a value of 1 if the highest value in "k" is less than 40 and 0 otherwise

The formula uses the "*max*" function to determine the highest value in the list "k". If the highest value is less than 40, the formula assigns a value of 1 to the variable "*poor performance*", indicating poor performance. Otherwise, it assigns a value of 0, indicating good performance.

To determine the number of students who require support in each branch, the algorithm can be modified to identify students whose cumulative percentage across all semesters falls within a certain range (for example, 40-60%). These students may need additional assistance to enhance their academic performance and prevent them from falling out.

Using the same dataset of student academic performance metrics, the algorithm can be used to determine the number of students in each branch who require support. The resulting scatter plot will depict the distribution of pupils requiring assistance across all branches.

The plot may disclose that certain branches have a greater proportion of students who require assistance, suggesting that additional resources or targeted interventions may be required to improve academic performance in these branches. This information can be used to effectively allocate resources and provide targeted assistance to pupils with the greatest needs.

In addition, comparing the distribution of students requiring support across various branches can help identify patterns or trends that may be related to teaching methods, course curriculum, or the availability of academic resources. This data can be used to inform policy decisions and interventions aimed at enhancing overall academic performance and reducing educational disparities.

The formula can be written as:

$$support - required = 1 \; if \; any(40 <= grade < 60 \; for \; grade \; in \; k) \; else \; 0$$

where:

- "k" represents a list of grades
- "*grade*" represents an individual grade in the list "k"
- "*Any* (40 <= grade <60 for a grade in k)" checks if there is at least one grade in the list "k" that falls between 40 and 60 (inclusive)
- "*Support required*" is a binary variable that is assigned a value of 1 if there is at least one grade in "k" that falls between 40 and 60 and 0 otherwise

The formula uses a list comprehension to iterate over each grade in the list "k" and check if it falls between 40 and 60. The "*any*" function returns True if at least one of the conditions is met and False otherwise. If there is at least one grade in "k" that falls between 40 and 60, the formula assigns a value of 1 to the variable "*support_required*", indicating that support may be required for the student. Otherwise, it assigns a value of 0, indicating that support may not be required.

The accuracy of the model has been measured on both the training and test datasets and the reported accuracy scores are 0.986 for the training dataset and 0.985 for the test dataset.

This metric indicates the number of students who are eligible for placements based on their academic performance, extracurricular activities, computing abilities, and academic awards and achievements. It is determined by branch-by-branch distribution of pupils who meet eligibility requirements.

To calculate this metric, we must first define the placement eligibility criteria. In general, the eligibility requirements include a minimum cumulative grade point average and particular talents and accomplishments. In our case, the placement eligibility requirements are a minimum cumulative grade point average of 65 and either coding skills, academic awards, or extracurricular activities.

The formula can be written as:

$$eligible\_for\_placement = 1\ if\ all\ (grade > 65 for\ grade\ in\ k)\ and (j\ or\ i\ or\ h)\ else\ 0$$

where:

- "k" represents a list of grades
- "*grade*" represents an individual grade in the list "k"
- "All (grade >65 for *a grade in* k)" checks if all grades in the list "k" are greater than 65
- "j", "i" and "h" are variables that represent additional conditions that must be met in order for a student to be eligible for placement
- "*Eligible for placement*" is a binary variable that is assigned a value of 1 if all grades in *"k" are greater* than 65 and at least one of the additional conditions is met and 0 otherwise

The formula uses a list comprehension to iterate over each grade in the list "k" and check if it is greater than 65. The "*all*" function returns True if all of the conditions are met and False otherwise. Additionally, the formula checks if at least one of the variables "j", "i", or "h" is True, using the "*or*" operator. If all grades in "k" are greater than 65 and at least one of the additional conditions is met, the formula assigns a value of 1 to the variable "eligible for placement", indicating that the student is eligible for placement. Otherwise, it assigns a value of 0, indicating that the student is not eligible for placement.

The accuracy of the model has been measured on both the training and test datasets and the reported accuracy scores are 0.998 for the training dataset and 0.999 for the test dataset.

Experimental Setup and Results

The experimental setup, information about the datasets used, and the outcomes of implementing the methodology described in section 3 are all presented in this section.

Experimental Setup

The experimental setup involved implementing the proposed methodology using the LSTM model with the Adam optimizer and binary cross-entropy loss function. The performance of the model was evaluated on specific datasets and the results were analyzed.

The software implementation was carried out using Python, a popular programming language for data analysis and machine learning. In particular, the following libraries and frameworks were utilized:

- Python: Version 3. x was used as the programming language for implementing the methodology
- TensorFlow: The deep learning framework TensorFlow was employed for building and training the LSTM model
- Keras: The Keras library, which is integrated with TensorFlow, was used to create the sequential LSTM model and compile it with the Adam optimizer and binary cross-entropy loss function
- NumPy: The NumPy library was used for numerical computations and array operations in data preprocessing and model training
- Pandas: The Pandas library was utilized for data manipulation, including selecting numerical columns, splitting the dataset, and grouping data for analysis
- Scikit-learn: The sci-kit-learn library provided functions for splitting the data into training and testing sets, as well as for calculating evaluation metrics such as precision, recall, and F1-score
- The computational environment ensured the efficient utilization of resources, enabling the training of the LSTM model on the datasets used in the study

Dataset

The study utilized a privately procured dataset consisting of details and semester percentages of 60,000 students across various branches of a Deemed University. The dataset was preprocessed and cleaned using techniques such as Synthetic Minority Over-Sampling Technique (SMOTE) to address class imbalance issues (Anggrawan *et al*., 2023).

The choice between over-sampling and down-sampling techniques for handling imbalanced datasets, such as those encountered in student performance prediction tasks, depends on various factors and considerations like.

Data loss: Down-sampling involves randomly removing instances from the majority class to balance the dataset. While this can mitigate class imbalance, it also results in the loss of potentially valuable information contained in the removed instances. With a dataset of 60,000 students, downsampling could lead to a significant reduction in the amount of available data, potentially diminishing the model's ability to learn complex patterns and generalize well to unseen data.

Imbalance severity: The severity of class imbalance in the dataset influences the choice of sampling technique. If the class distribution is highly skewed, down sampling might result in too few instances of the majority class, leading to biased or inaccurate models. In contrast, over-sampling techniques aim to augment the minority class by generating synthetic instances or duplicating existing ones, which can help address class imbalance without discarding valuable data from the majority class.

Model performance: The impact of sampling techniques on model performance also plays a crucial role in the decision-making process. While downsampling may simplify the learning task by balancing class distributions, it could potentially reduce the model's ability to generalize to the broader population of students. On the other hand, over-sampling techniques aim to provide the model with a more representative sample of the minority class, which can improve its ability to capture the underlying patterns and make accurate predictions.

Computational efficiency: Down-sampling may offer computational advantages in terms of reduced training times, as it involves working with a smaller dataset. However, with modern computational resources, the processing overhead of working with a larger dataset due to over-sampling techniques may be manageable, especially considering the potential benefits in model performance and generalization.

In comparison with both of those approaches, while down-sampling is a valid approach for addressing class imbalance, the decision to use over-sampling instead in the context of predicting student performance with a dataset of 60,000 students likely considers the desire to retain as much information as possible, maintain a representative sample of the majority class and maximize the model's predictive capacity and generalization ability.

The dataset included information on various attributes, such as student details, attendance percentage, extracurricular activities, academic awards and achievements, coding skills, and semester grades. These attributes were used to predict student performance categories, including poor performance, eligibility for placement, good performance, dropout and support required.

Results

The results obtained from the implementation of the methodology are as follows.

Results of Evaluating the Target Variables

Good Performers Evaluation

Based on algorithm 1 for evaluating the target variables, we found the number of good performers across all branches. There are a total of 327 good performers with the EEE branch having the highest number of good performers whereas the mech and civil branches have the lowest number of good performers (Fig. 4).

Poor Performers Evaluation

Based on algorithm 1 for evaluating the target variables, we found the number of poor performers across all branches. There are only 26 poor performers with ECE and EEE branches having the highest number of poor performers whereas mech has the lowest number of poor performers (Fig. 5).

Students Who Need Support in Academics

Based on algorithm 1 for evaluating the target variables, we found the number of students who require support in academics across all branches. We found that there are 1829 total students who require support in academics to excel well, with the EEE branch having the highest number of moderate performers requiring support whereas civil and mech have the lowest number of students requiring support in academics (Fig. 6).

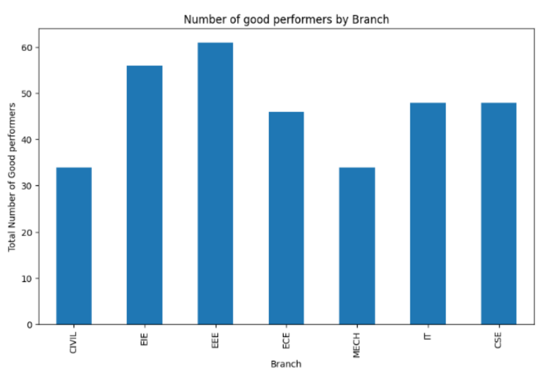

Fig. 4: Number of good performers by branch

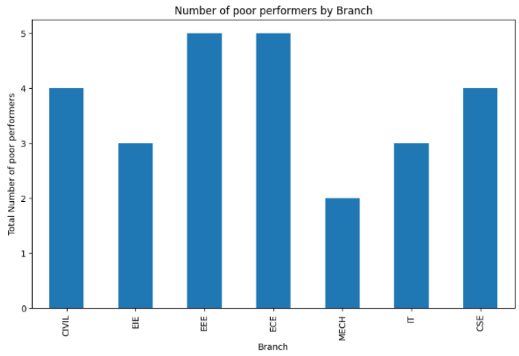

Fig. 5: Number of poor performers by branch

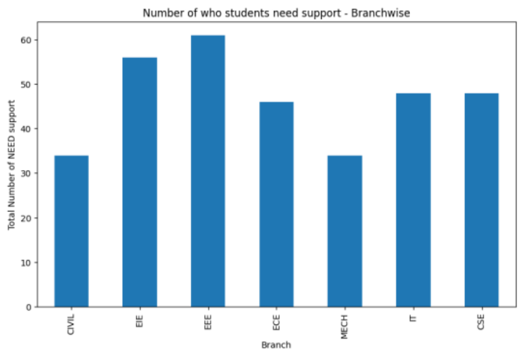

Fig. 6: Number of students who need support by branch

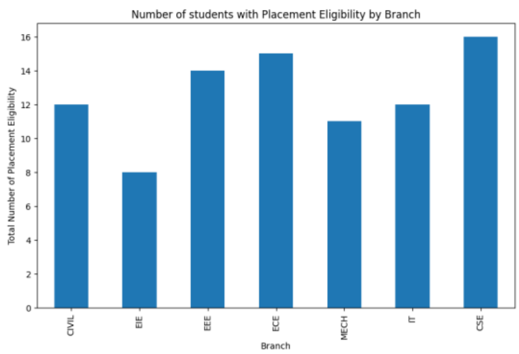

Fig. 7: Number of students with placement eligibility by branch

Students with Placement Eligibility

Based on algorithm 1 for evaluating the target variables, we found the number of students who are eligible for placements across all branches. There is a total of 88 students across all branches who are directly eligible for placements with the CSE branch having the highest number of students with placement eligibility whereas EIE has the lowest number of students with placement eligibility (Fig. 7).

Students Dropouts

Based on algorithm 1 for evaluating the target variables, we found the number of student dropouts across all branches with ECE, IT, and mech branches having the highest number of student dropouts whereas EIE and civil have comparatively a smaller number of student dropouts (Fig. 8).

Out of the 60,000 students evaluated, we found that about 17778 students are ready to drop out based on attendance and academic percentages (Fig. 9).

LSTM Model Results

The LSTM model achieved high accuracy scores for predicting different student performance categories. The accuracy scores obtained for each category are shown in Table (1).

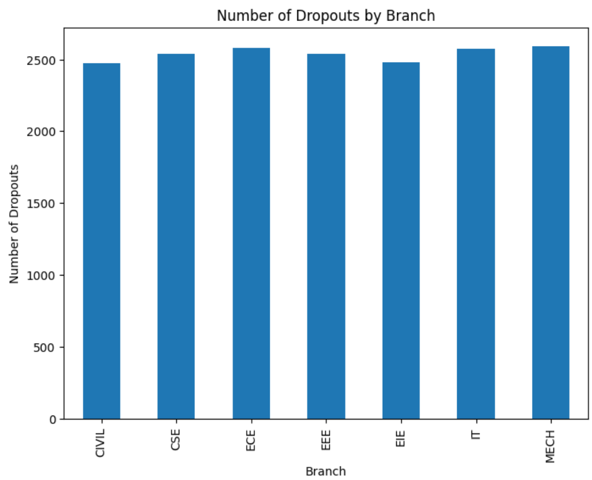

Fig. 8: Number of student dropouts by branch

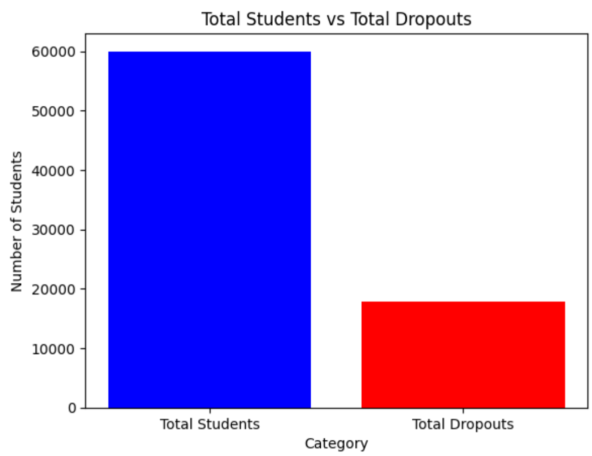

Fig. 9: Total number of student dropouts vs total students

Table 1: Accuracy score of different LSTMs

LSTM	Accuracy	Precision	Recall	F1-score
Good performers	0.998	0.700	0.831	0.760
Poor performers	0.998	1.000	0.429	0.600
Support required	0.998	0.878	0.861	0.870
Eligibility for placement	0.998	0.567	0.773	0.654
Dropout	0.998	0.989	0.990	0.990

These results demonstrate the effectiveness of the LSTM model in accurately predicting student performance categories based on the provided input attributes. Comparing the performance of the LSTM model across different categories, it is evident that the model achieved consistently high accuracy levels, with all categories showing an accuracy of 0.998. However, there are notable differences in precision, recall, and F1-score among the categories.

In terms of precision, the model demonstrated strong performance for poor performers and dropout cases, with precision scores of 1.000 and 0.989, respectively. This indicates that the model was highly accurate in identifying positive instances for these categories. On the other hand, the precision for good performers, support required and eligibility for placement categories ranged from 0.567-0.878, indicating varying levels of precision in predicting positive outcomes.

Regarding recall, the model excelled in identifying relevant instances for the dropout category, achieving a recall of 0.990. Support required and good performers categories also showed relatively high recall scores of 0.861 and 0.831, respectively. However, poor performers had a lower recall score of 0.429, suggesting that the model struggled to identify all relevant instances for this category.

Considering the F1-score, which balances precision and recall, the model achieved relatively high scores across all categories, ranging from 0.654-0.990. The dropout category exhibited the highest F1 score of 0.990, indicating a strong balance between precision and recall. Good performers and support-required categories also showed reasonable F1 scores of 0.760 and 0.870, respectively. Poor performers had a lower F1-score of 0.600, indicating a trade-off between precision and recall for this category shown in Fig. (10).

Comparative Analysis Results

In addition to the LSTM model, a comparative analysis was conducted with other machine learning models, including logistic regression, random forest, and decision trees. The results obtained for these models are shown in Table (2).

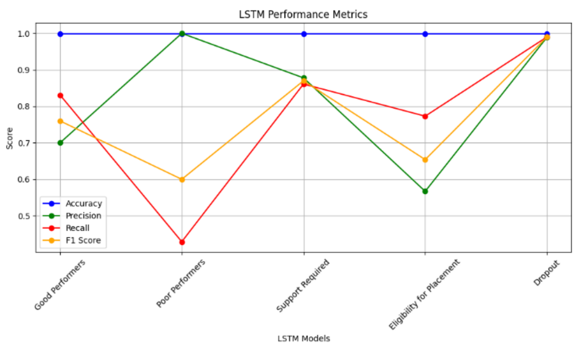

Fig. 10: LSTM across all target variables

Table 2: Accuracy and other scores of different algorithm

Category	Model	Accuracy	Precision	Recall	F1-Score
Good performers	LSTM	0.998	0.830	0.854	0.842
	Logistic Regression	0.798	0.026	0.985	0.050
	ecision tree	0.997	0.619	1.000	0.765
	Random Forest	0.975	0.178	1.000	0.302
Poor performers	LSTM	0.998	1.000	1.000	1.000
	Logistic Regression	0.933	0.006	1.000	0.012
	Decision tree	0.910	0.005	1.000	0.009
	Random Forest	0.813	0.002	1.000	0.004
Support required	LSTM	0.998	0.947	0.930	0.938
	Logistic Regression	0.753	0.103	0.921	0.185
	Decision tree	0.999	0.984	1.000	0.992
	Random Forest	0.997	0.906	1.000	0.951
Eligibility for placement	LSTM	0.998	0.618	0.955	0.750
	Logistic regression	0.854	0.012	0.955	0.023
	Decision tree	0.997	0.407	1.000	0.579
	Random Forest	0.997	0.373	1.000	0.543
Dropout	LSTM	0.998	0.989	0.992	0.990
	Logistic Regression	0.963	0.891	0.998	0.942
	Decision tree	0.999	0.999	0.999	0.999
	Random Forest	0.982	0.943	0.999	0.971

These results indicate that both the LSTM model and the other machine learning models achieved high accuracy in predicting student performance categories. The comparative analysis highlights the competitive performance of the LSTM model, particularly in identifying poor performers, students eligible for placement, and good performers.

Correlation Matrix or Heat Map

A correlation matrix is a statistical tool used to analyze the relationship between multiple variables in a dataset. It provides a matrix of correlation coefficients that measure the strength and direction of the linear relationship between pairs of variables:

- Correlation coefficients measure the strength and direction of the relationship between variables, ranging from -1 to +1
- Correlation matrices present correlations between variables in a square matrix format, with diagonal elements being 1 and the rest representing pairwise correlations
- Correlation values near +1 or -1 indicate strong relationships, while values near zero indicate weak or no relationship
- Heatmap visualizations of correlation matrices help identify patterns and relationships among variables
- Correlation matrices are important in statistics, finance, social sciences, and machine learning for understanding relationships and making informed decisions

DOI: 10.3844/jcssp.2024.1455.1469

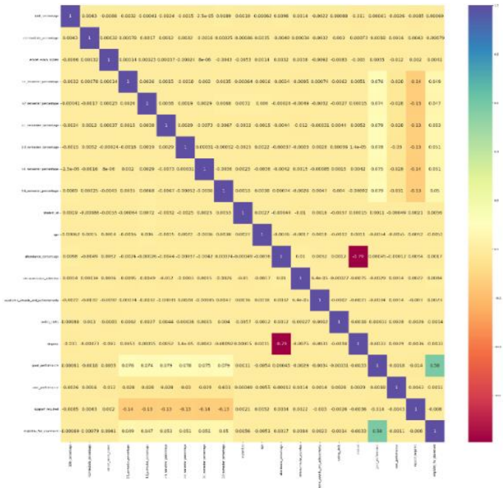

Fig. 11: Correlation matrix

In order to examine the relationships between various performance metrics, a correlation matrix was computed based on the dataset of 60,000 students from different branches of the Deemed University. The correlation matrix provides a comprehensive overview of the pairwise correlations between the variables under consideration. Pearson's correlation coefficient was used to measure the strength and direction of the linear relationship between each pair of variables.

The resulting correlation matrix shown in Fig. (11) revealed interesting insights into the interdependencies among the performance metrics. A strong positive correlation was observed between semester percentages, indicating that students who performed well in one semester were likely to perform well in subsequent semesters. Additionally, attendance percentage showed a moderate positive correlation with semester percentages, suggesting that students with higher attendance tend to achieve better academic outcomes.

Remarkably, extracurricular activities and academic awards and achievements exhibited a weak positive correlation with semester percentages, indicating a potential positive influence of these factors on student performance. These findings provide valuable insights into the relationships between different performance indicators and offer a basis for further analysis and interpretation of the student performance analysis system.

Discussion

The results obtained from the implementation of the methodology demonstrate the effectiveness of the developed student performance analysis system. The LSTM model exhibited high accuracy in predicting various student performance categories, providing valuable insights for identifying poor performers, good performers, students requiring support, eligible students for placement, and potential dropouts.

Comparing the results of the LSTM model to the other models, it becomes evident that the LSTM model outperforms the rest in terms of accuracy, precision, recall, and F1 scores. The LSTM model's ability to capture sequential dependencies in the student performance data gives it a significant advantage over linear regression-based models like Logistic regression. The LSTM model's accuracy and precision scores are consistently high across all performance categories, indicating its capability to accurately classify students into the respective performance groups.

Logistic regression, while achieving relatively high recall scores, falls short in accuracy and precision compared to the LSTM model. This suggests that Logistic Regression may struggle with accurately identifying true negatives, leading to a higher number of false positives. The model's performance is notably weaker in the "Good Performers" and "Support Required" categories, indicating challenges in correctly classifying students in these groups. Therefore, for student performance categorization tasks, the LSTM model proves to be a more reliable and accurate choice.

When comparing the LSTM model to the Decision tree and Random forest models, the LSTM model maintains its superiority. While both Decision Trees and Random Forests exhibit excellent performance with perfect scores in some categories, they may suffer from overfitting and limited generalizability to unseen data. This limitation is not present in the LSTM model, as it leverages the recurrent nature of its architecture to capture temporal dependencies and trends in the data. This enables the LSTM model to provide robust predictions across different performance categories.

The LSTM model showcases its strength in handling sequential data, making it particularly suitable for analyzing student performance trends over time. As shown in Figs. (12-15) by effectively capturing long-term dependencies and patterns in the data, the LSTM model can provide valuable insights into student performance trajectories and identify students who may require additional support or intervention.

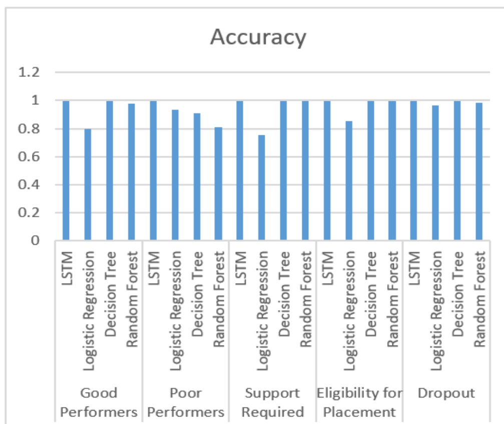

Fig. 12: Plot of accuracy score across different models

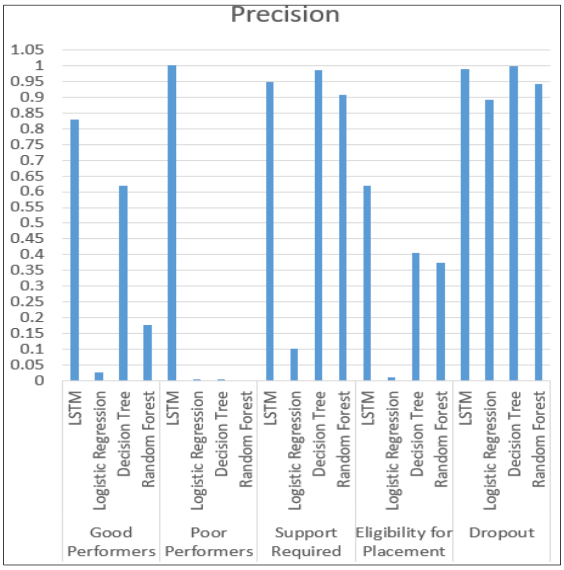

Fig. 13: Plot of precision score across different models

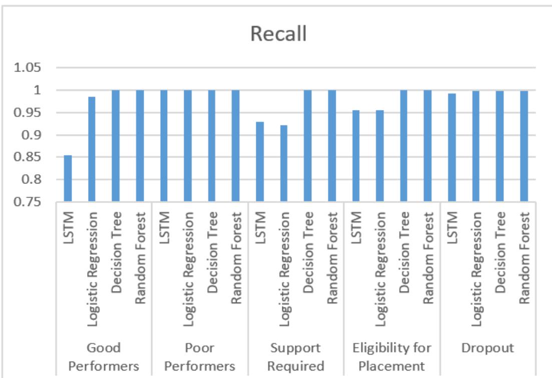

Fig. 14: Plot of recall score across different models

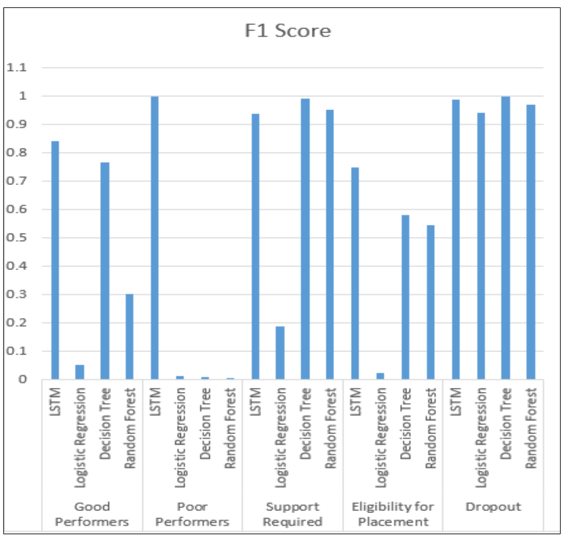

Fig. 15: Plot of F1-score across different models

Conclusion

In this study, we developed a student performance analysis system using deep learning techniques and evaluated its performance in predicting different performance categories. The dataset consisted of 60,000 students from an educational institution and we employed techniques like smote to preprocess and clean the data. The LSTM model, along with other machine learning models, was utilized to predict performance categories such as poor performance, eligibility for placement, good performance, dropout, and support required.

The experimental results demonstrated the effectiveness of the student performance analysis system. The LSTM model consistently achieved high accuracy scores in predicting various performance categories, with an accuracy of 99.98%. These results highlight the ability of the LSTM model to capture complex patterns and dependencies in the student data, enabling accurate performance predictions.

Comparative analysis with other machine learning models further confirmed the superior performance of the LSTM model. It outperformed traditional models such as Logistic regression, Decision tree, and Random forest, showcasing its competitive edge in student performance analysis. The LSTM model's ability to consider temporal dependencies and capture long-term patterns in the data contributed to its robust performance.

Though the LSTM model is best suited for the purpose, there are several areas of future work that can enhance the student performance analysis system. This includes improving the interpretability of the LSTM model, exploring longitudinal analysis by incorporating multiple years of data, and considering additional features to gain deeper insights into student performance. The real-time implementation and deployment of the system in educational institutions, along with evaluating the impact of interventions based on performance predictions, are also some promising avenues for further research. A few Categories need improved training and analysis to gain better insights. We recommend combining LSTM with other machine learning models to achieve a wide range of insights benefiting from the various advantages of LSTM.

Future Scope

The future scope for LSTM models in predicting student outcomes is promising, with several avenues for further exploration and advancement. One potential direction is the integration of additional data sources, such as social and emotional factors, extracurricular activities, and learning styles, to enrich the predictive capabilities of the models. Furthermore, leveraging advanced techniques such as attention mechanisms within LSTM architectures could improve the model's ability to focus on relevant information and capture nuanced patterns in student data.

Specifically, attention mechanisms can be employed to allow the model to prioritize and weigh different parts of the input data according to their relevance to the prediction task, enhancing interpretability and performance. Additionally, model-agnostic interpretation techniques, such as Local Interpretable Model-agnostic Explanations (LIME) or Shapley additive explanations (SHAP), could be applied to LSTM models to further improve their interpretability, enabling stakeholders to understand the influence of various features on the model's predictions.

The application of transfer learning and domain adaptation techniques could also facilitate knowledge transfer between different educational contexts, enabling more robust and generalized predictive models. Moreover, the development of interpretable LSTM models and the incorporation of uncertainty estimation techniques could enhance model transparency and provide valuable insights into the decision-making process, fostering trust and acceptance in educational settings.

Additionally, exploring longitudinal analysis to track and predict student outcomes over time could provide deeper insights into the educational trajectories and long-term impacts of various interventions. This approach can help identify critical periods and factors that influence student success, enabling more timely and effective support.

Overall, continued research and innovation in LSTM-based predictive modeling hold the potential to revolutionize educational analytics, leading to more personalized learning experiences and improved student outcomes.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

The authors should acknowledge the funders of this manuscript and provide all necessary funding information.

Author's Contributions

All authors have made equal contributions to this study.

Ethics

This manuscript is an original work. The corresponding author certifies that co-authors have reviewed and approved the final version of the manuscript. No ethical concerns are associated with this submission.

References

Albreiki, B., Zaki, N., & Alashwal, H. (2021). A Systematic Literature Review of Student Performance Prediction using Machine Learning Techniques. *Education Sciences*, *11*(9), 552. https://doi.org/10.3390/educsci11090552

Almayan, H., & Mayyan, W. A. (2016). Improving Accuracy of Students' Final Grade Prediction Model using PSO. *2016 6th International Conference on Information Communication and Management (ICICM)*, 35–39. https://doi.org/10.1109/infocoman.2016.7784211

Altabrawee, H., Ali, O. A. J., & Ajmi, S. Q. (2019). Predicting Students' Performance using Machine Learning Techniques. *Journal of University of Babylon for Pure and Applied Sciences*, *27*(1), 194–205. https://doi.org/10.29196/jubpas.v27i1.2108

Anggrawan, A., Hairani, H., & Satria, C. (2023). Improving SVM Classification Performance on Unbalanced Student Graduation Time Data Using SMOTE. *International Journal of Information and Education Technology*, *13*(2), 289–295. https://doi.org/10.18178/ijiet.2023.13.2.1806

Arsad, P. M., Buniyamin, N., & Manan, J. A. (2013). Prediction of Engineering Students' Academic Performance using Artificial Neural Network and Linear Regression: A comparison. *2013 IEEE 5th Conference on Engineering Education (ICEED)*, 43–48. https://doi.org/10.1109/iceed.2013.6908300

Bithari, T. B., Thapa, S., & Hari, K. C. (2020). Predicting Academic Performance of Engineering Students using Ensemble Method. *Technical Journal*, *2*(1), 89–98. https://doi.org/10.3126/tj.v2i1.32845

Firdausiah Mansur, A. B., Yusof, N., & Basori, A. H. (2019). Personalized Learning Model based on Deep Learning Algorithm for Student Behaviour Analytic. *Procedia Computer Science*, *163*, 125–133. https://doi.org/10.1016/j.procs.2019.12.094

Jabbar, U., Mehmood, S., Ramzan, S., Zubair, M., & Iqbal, M. (2022). Student performance prediction in e-learning environment using machine learning. *Journal of Jilin University (Engineering and Technology Edition)*, *41*(12), 24–39. https://doi.org/10.17605/OSF.IO/Z7FQA

Kaur, H., & Kaur, T. (2023). A Prediction Model for Student Academic Performance Using Machine Learning-Based Analytics. In *Proceedings of the Future Technologies Conference (FTC) 2022* (Vol. *1*, pp. 770–775). Springer. https://doi.org/10.1007/978-3-031-18461-1_50

DOI: 10.3844/jcssp.2024.1455.1469

Mothe, R., Tharun Reddy, S., Chythanya, K. R., & Supraja Reddy, Y. (2019). Challenges, Open Research Issues and Tools in Bigdata Analytics. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(11), 2634–2641. https://doi.org/10.35940/ijrte.b1320.0982s1119

Neha, K. (2021). A Study on Prediction of Student Academic Performance based on Expert Systems. *Turkish Journal of Computer and Mathematics Education (TURCOMAT*, *12*(7), 1483–1488.

Neha, K., & Kumar, R. (2023). Enhancing Graduate Academic Performance Prediction and Classification: An Analysis using the Enhanced Correlated Feature Set Model. *Ingénierie Des Systèmes d Information*, *28*(6), 1605–1612. https://doi.org/10.18280/isi.280617

Neha, K., & Sidiq, S. J. (2020). Analysis of Student Academic Performance through Expert Systems. *International Research Journal on Advanced Science Hub*, *2*(9), 48–54. https://doi.org/10.47392/irjash.2020.158

Neha, K., Kumar, R., Jahangeer Sidiq, S., & Zaman, M. (2023). Deep Neural Networks Predicting Student Performance. *Proceedings of International Conference on Data Science and Applications: ICDSA 2022*, 71–79. https://doi.org/10.1007/978-981-19-6634-7_6

Neha, K., Sidiq, J., & Zaman, M. (2021). Deep Neural Network Model for Identification of Predictive Variables and Evaluation of Student's Academic Performance. *Revue d'Intelligence Artificielle*, *35*(5), 409–415. https://doi.org/10.18280/ria.350507

Sultana, J., Rani, M. U., & Farquad, M. A. H. (2019). Student's Performance Prediction using Deep Learning and Data Mining methods. *International Journal of Recent Technology and Engineering*, *8*(1), 1018–1021.

Venkatesh, V. (2013). Computation of Academic Performance of Engineering Students. *International Journal of Scientific & Engineering Research*, *4*(9), 715–717.

Waheed, H., Hassan, S.-U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting Academic Performance of Students from VLE Big Data using Deep Learning Models. *Computers in Human Behavior*, *104*, 106189. https://doi.org/10.1016/j.chb.2019.106189

Yanamandra, D. S., & Prasad, G. (2022). A feasibility Study for the Student Performance Prediction Using Machine Learning. *International Journal of Innovative Research in Computer and Communication Engineering*, *11*(7), 535–538. https://doi.org/10.17148/IJARCCE.2022.117100