IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2023, accepted 29 December 2023, date of publication 8 January 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351186

RESEARCH ARTICLE

Prediction of Course Grades in Computer Science Higher Education Program via a Combination of Loss Functions in LSTM Model

ANAHITA GHAZVINI[1], **(Member, IEEE),**
NURFADHLINA MOHD SHAREF[1,2], **(Senior Member, IEEE),**
AND FATIMAH BINTI SIDI[3], **(Member, IEEE)**

[1] Intelligent Computing Research Group, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia
[2] Institute of Mathematical Research, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia
[3] Database Technology and Applications Group, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia

Corresponding author: Nurfadhlina Mohd Sharef (nurfadhlina@upm.edu.my)

This work was supported by the project "Interactive Machine Learning based on Deep Reinforcement Learning and Generative Adversarial Network Hybrid for Digital Twin" under Grant FA2386-21-1-4050.

ABSTRACT In the realm of education, the timely identification of potential challenges, such as learning difficulties leading to dropout risks, and the facilitation of personalized learning, emphasizes the crucial importance of early grade prediction. This study seeks to connect predictive modeling with educational outcomes, particularly focusing on addressing these challenges in computer science higher education programs. To address these issues, nonlinear dynamic systems, notably Recurrent Neural Networks (RNNs), have demonstrated efficacy in unraveling the intricate relationships within student learning traces, surpassing the constraints of traditional time series methods. However, the challenge of vanishing gradient issues hampers RNNs, leading to a significant decrease in gradient values during weight matrix multiplication. To solve this challenge, we introduce an innovative loss function, the MSECosine loss function crafted by seamlessly combining two established loss functions: Mean Square Error (MSE) and LogCosh. In assessing the performance of this novel loss function, we employed two self-collected datasets comprising learning management system (LMS) and assessment records from a higher education computer science program. These datasets serve as the testing ground for four deep time series models: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory network (LSTM), and CNN-LSTM. Employing 29 meticulously designed feature sets representing combination of demography, learning activities and assessment, LSTM emerges as the preeminent model which is consistent with our expectation that RNN is the best suited approach. Building on this groundwork, we solve the vanishing gradient issue and boost the LSTM model's performance by integrating the proposed MSECosine loss function, resulting in an enhanced model termed eLSTM. Experimental results underscore the noteworthy achievements of the eLSTM model, emphasizing an accuracy of 0.6191% and a substantially reduced error rate of 0.1738. The proposed MSECosine loss function performance in addressing the vanishing gradient issue yields two times better than compared to standard loss functions. These outcomes surpass those of alternative approaches, highlighting the instrumental role of the MSECosine loss function in refining eLSTM for more accurate predictions in course grade prediction, as well as the feature set that captures early grade prediction.

INDEX TERMS Learning analytics, hybrid objective functions, deep time series neural network, LSTM model optimization.

I. INTRODUCTION

Detecting students who are at risk of dropping out or failing is important, and predicting their academic performance early can be highly beneficial [1], [2], and [3]. Academic performance is the main factor in evaluating the quality of education for college students [4], [5], [6], [7], and [8]. Therefore, the early grade prediction can be achieved through the use of sequential data that contains previous information about the student's activities [9], [10]. Thus, to deal with this type

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

of data, dynamic systems have to be employed. One way to analyze dynamic systems is by the usage of time series approaches [11].

Accordingly, various time series forecasting schemes, such as simple, autoregressive, and exponential smoothing approaches, have been used in the past for early prediction in different fields e.g., economics, stock market, and engineering [11], [12], and [13]. However, these methods are limited in their ability to learn complex patterns and can only handle simple prediction challenges using linear methods. Also, they demand a significant quantity of data to attain a high level of accuracy [11], [12], and [13].

Deep learning models can address this limitation by being employed for time series forecasting tasks [14]. Deep time series models by leveraging the power of neural networks can effectively handle non-linear relationships and capture intricate patterns that may be missed by linear methods [14]. Consequently, this ability makes the deep time series scheme to be well-suited for challenging prediction tasks such as grade prediction [14], [15].

This study specifically addresses the prediction of course grades in a computer science higher education program via a novel MSECosine of loss functions in the LSTM model. While traditional time series forecasting schemes, including simple, autoregressive, and exponential smoothing approaches, have been employed in various fields for early prediction [11], [12], and [13], they exhibit limitations in learning complex patterns and are restricted to linear methods.

Deep learning methods such as Multilayer perceptron (MLP), Convolutional neural network (CNN), Long Short-Term Memory (LSTM) [16], [17] and combination of CNN and LSTM method (CNN-LSTM), known as the powerful alternatives [16], [18]. However, these methods suffer from the issue of vanishing and exploding gradients during the training phase [16], [18].

The existing literature on early grade prediction using these deep learning methods has primarily focused on understanding temporal relationships through new gating techniques, employing LSTM [16], [18], and [19]. However, this body of work has not thoroughly explored the significant issue of error magnification during the training phase [16], [18], and [19].

Accordingly, one way to address this issue is by utilizing suitable loss functions, such as Mean Square Error (MSE), which is considered the best function as it does not suffer from vanishing or exploding gradients due to its use of an exponential term [16], [18]. However, the MSE function can magnify errors when the network is not performing well [16], [18]. To tackle this problem, the logarithm function can be used to prevent the exponential function from expansion, reducing the skewness of exponential terms [17].

Hence, this study aims to fill this gap between predictive modeling and educational outcomes by proposing a novel loss function, MSECosine, to address this specific challenge and enhance network performance. This innovative loss function, a combination of Mean Square Error (MSE) and LogCosh, aims to address the vanishing gradient problem, presenting a unique approach that distinguishes our work from previous efforts.

Our hypothesis posits that the proposed MSECosine loss function effectively addresses the issue of error magnification during the training phase, leading to improved performance in early grade prediction. To investigate this, our research question examines how the MSECosine loss function impacts the performance of deep time series models in predicting student grades early on.

If the combination of the logarithm function can prevent the exponential loss function from growing, then combining the Logcosh and MSE loss functions will produce a desirable network error. In this research, we aim to suggest the novel MSECosine loss function to address error magnification during the training phase by combining the MSE and Logcosh loss functions.

The contribution of this work is threefold. Firstly, we present the results of an early grade prediction paper through an empirical investigation of the accuracy of four deep time series models. Secondly, we propose a method called eLSTM, introducing the new loss function MSECosine as a solution to the vanishing gradient problem in deep time series models. Thirdly, we present the optimized early grade prediction technique using the proposed eLSTM based on the evaluation of 29 feature sets.

The remaining of this paper is segmented into five sections. The literature review of the essential of early grade prediction using deep time series models as well as the impact of loss function to the network performance are covered in Part II. The methodology is discussed in Part III. The enhancement of the LSTM model using proposed MSECosine loss function is addressed in Part V. Attained results and the discussion are mentioned in Part V. The discussion section is stated in Part VI. Conclusively, the paper is concluded in Part VII.

II. RELATED WORK

This section discusses the existing literature on time series prediction within the context of academic performance forecasting, emphasizing the limitations of traditional time series methods and the advantages of deep learning techniques, specifically Recurrent Neural Networks (RNNs) like LSTM.

Additionally, it highlights the superiority of LSTM over other models in predicting student performance, as demonstrated in various studies. Furthermore, we address the limitations of these models, which are overcome through the use of appropriate components such as loss functions.

Numerous optimization algorithms are currently being applied across diverse academic disciplines, including mechanical engineering, automobile engineering, aerospace engineering, etc. The relevance of these studies becomes particularly significant when considering the dynamic nature of academic performance [20], [21].

Academic performance, a nonlinear dynamic system, exhibits feedback loops, chaotic behavior, and sensitivity to initial conditions. Unlike linear systems, where input-output relationships are straightforward, nonlinear systems can have complex and disproportionate connections. This complexity is further compounded by the dynamic nature of academic performance, subject to changes over time due to various influences like study habits, motivation, and external factors.

Such intricate relationships mean that minor alterations in input variables can trigger significant and unpredictable shifts in output. The multifaceted interplay of factors affecting academic performance defies linear models. The evolution of these influences over time introduces volatility, potentially leading to unexpected fluctuations or abrupt changes in academic outcomes. For instance, doubling study hours may not result in double grade improvement, due to intricate variable interactions and diminishing returns.

Recognizing the significance of nonlinear dynamic systems in academic performance prediction, deep time series models offer advantages over traditional methods. Nonlinear techniques are vital for predicting time delays inherent in dynamic systems, often encountered in time series forecasting [18], [22].

In various domains, researchers have employed traditional time series techniques, ranging from basic methods like averaging to more complex approaches such as nonlinear autoregressive networks with exogenous inputs (NARX) [23] and exponential smoothing (Figure. 1) [23], [24]. These methods address prediction challenges but may fall short in capturing the intricate dynamics of academic performance.

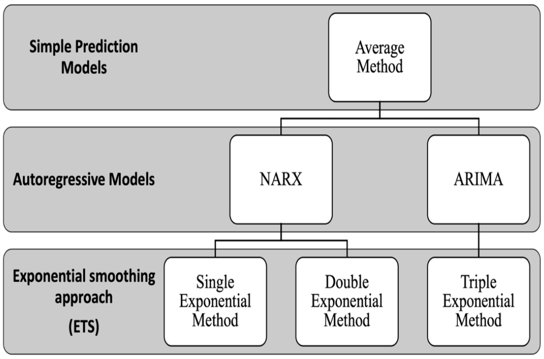

Simple Prediction Models
Average Method
Autoregressive Models
NARX
ARIMA
Exponential smoothing approach (ETS)
Single Exponential Method
Double Exponential Method
Triple Exponential Method

FIGURE 1. Traditional time series approaches.

Classic time series models (Figure. 2) rely solely on past inputs and current histories for predictions. While effective in industries like sales estimation, energy consumption forecasting, and passenger predictions, their linear nature limits their ability to handle complex patterns and latent factors [22]. These linear constraints lead to vulnerability to outliers, time inefficiency, and data limitations, necessitating heuristics and fine-tuning, especially for seasonality [22].

In contrast, deep time series methods, such as Recurrent Neural Network (RNN), have demonstrated remarkable performance across various domains, capturing intricate patterns and relationships [23], [24]. They provide more accurate control and forecasting in partially unknown environments, as evidenced by their success in natural language processing, image classification, and more [25], [26], and [27].

As a result, various deep learning techniques, including MLP, CNN, LSTM, and CNN-LSTM combinations, have been employed for time series prediction [13], [28]. These methods excel in capturing complex time-dependent relationships, as highlighted in Table 1.

TABLE 1. Application of deep learning techniques for time series prediction.

Model	Advantage	Author
MLP	• Robust to Noise • Nonlinear • Multivariate Inputs • Multi-step Forecasts	[29]
CNN	• Automatic identification • Automatic extraction • Support several variables as input and output • Acquiring intricate and arbitrary functional relationships • Mastering extensive input sequences vital for prediction tasks	[30], [31]
LSTM	• Native Support for Sequences • Learned Temporal Dependence	[18]

Academic performance forecasting is commonly achieved through grade prediction using standard machine learning (e.g., MLP, SVM) and time series (e.g., LSTM, GRU) methods [29], [32]. Notably, LSTM consistently outperforms other models in terms of precision, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) due to its effective capture of complex temporal dependencies in educational data [1]. See Table 2 for a comprehensive overview of these approaches.

The mentioned table demonstrates LSTM's efficacy in predicting student performance by surpassing baseline models, such as SVM and MLP, with superior accuracy [29]. LSTM's adeptness in handling sequential data and retaining long-term information enhances its adaptability to the complexities of educational data, resulting in enhanced predictions [28]. In contrast to alternatives, LSTM offers reliable and precise predictions, making it the preferred model for educational grade prediction [18].

Comparing Artificial Neural Network (ANN) with RNN and Bidirectional Long Short-Term Memory (Bi-LSTM), the study finds that Bi-LSTM excels in predicting students' final total scores [18], [31]. Bi-LSTM's combination with LSTM and bidirectional architecture augments its understanding of sequential data, consistently outperforming ANN and RNN [18].

These comparisons underscore LSTM's and Bi-LSTM's superiority in predicting student performance. Their ability to process sequential data and comprehend temporal

TABLE 2. Application of deep learning techniques for time series prediction. Standard and deep time series approaches for grade prediction.

Dataset Info/year	Model	Top Model	Metrics	Author
Student info from graduates' program (2014-2018)	1. MLP 2. Naïve Bayes 3. IBk 4. J48	MLP	-Accuracy -RMSE	[29]
Students grades in a certain school (Not stated)	1. SVM 2. MLP	LSTM	-MAE -RMSE -MSE	[29]
Open University Learning Analytics Dataset (OULAD) (2013-2014)	None	LSTM	-Precision -Accuracy -F1-Score -Recall	[18]
Students' performance in higher education (2007-2019)	1. DNN 2. LR	LSTM	-Mean Absolute -Error (MAE) -RMSE -Accuracy	[16],[34]
Collected from a multidisciplinary university (2009-2019)	None	-RNN -Bi-LSTM	-Accuracy	[18]

dependencies proves promising for accurate and effective grade prediction. While both LSTM and Bi-LSTM [18], [33] exhibit strong performance, LSTM's computational efficiency sets it apart. Operating unidirectionally, LSTM processes input sequences from past to future or vice versa, reducing complexity and memory needs compared to Bi-LSTM [16]. Consequently, the LSTM model [34], [35], [36], [37] emerges as a compelling choice for grade prediction in educational data analysis, offering a balance of performance and computational efficiency.

Although the above-mentioned approaches are showing promising results for time series prediction, these methods suffer from the issue of vanishing gradient due to working on large-scale time series forecasting problems. Several loss functions can overcome this issue. The above-mentioned deep time series forecasting methods are applied into a regression-type predictive modeling problem [33]. Multiple regression loss functions have been proposed to improve the performance evaluation in regression analysis. The advantage and disadvantage of the regression loss function is specified in Table 3.

Among the above presented loss functions, MSE and LogCosh loss functions are considered as the most well-known regression functions. This is due to their differentiability that avoids vanishing gradient issues while using large sequential data. The MSE differentiability makes this function easily accomplish the mathematical operations. Also, this function evaluates the fitness of the model by producing lower value.

TABLE 3. Pros and cons of regression loss functions.

Model	Advantage	Disadvantage	Author
Mean Absolute Error (MAE)	-Computationally cheap -Not sensitive to outlier	-Suffer from vanishing gradient issue due to absolute terms	[32]
Mean Squared Error (MSE)	-Not suffer from vanishing gradient issue -Differentiable	-Magnifying error value once the network performs poor -Sensitive to outlier	[31]
Mean Bias Error (MBE)	-Useful to identify and correct model bias	-Not applicable for numbers ranging from $(-\infty, \infty)$	[33]
LogCosh Loss	-Differentiable -Required few computation	-Less adaptable since it operates on a fixed scale	[13]

Whilst, the LogCosh function is beneficial for keeping balance as it utilizes the logarithm terms.

III. MATERIAL AND METHODS

This section outlines the methodology used in this research, which is divided into five phases of 1) Data Collection, 2) Implementation of various features selection models, 3) Modeling of predictive deep time series models, 4) Evaluation metrics that applied in all tested models, and 5) Enhancement of best model using the proposed MSECosine loss function. Figure. 2 represents the sequence of the methodology of this work.

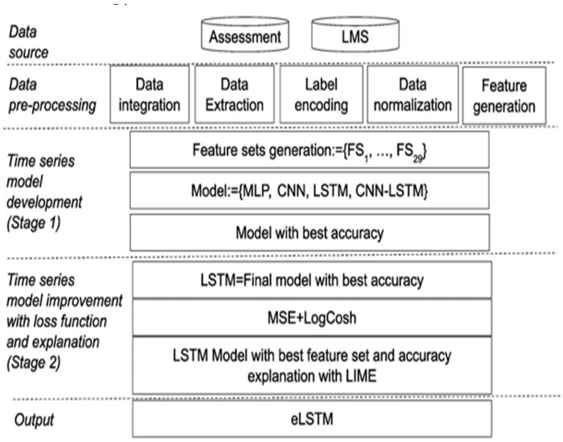

Data source
Assessment
LMS
Data pre-processing
Data integration
Data Extraction
Label encoding
Data normalization
Feature generation
Time series model development (Stage 1)
Feature sets generation:={FS$_1$, ..., FS$_{29}$}
Model:={MLP, CNN, LSTM, CNN-LSTM}
Model with best accuracy
Time series model improvement with loss function and explanation (Stage 2)
LSTM=Final model with best accuracy
MSE+LogCosh
LSTM Model with best feature set and accuracy explanation with LIME
Output
eLSTM

FIGURE 2. The research methodology phase.

The data collection and preparation phase provided the information on 1) Description of self-collected dataset of 'LMS' and 'Assessment', 2) Procedure of merging these two datasets, and 3) Preprocessing methods. The second phase presents the details on implementation of 29 designed feature selection and their importance. Furthermore, the third phase displays the principle and proposed framework of this study using four different time series models of MLP, CNN, LSTM, and CNN-LSTM. Afterwards, phase fourth specified the evaluation metrics that applied to quantitatively assess the performance of each of the tested methods by this study. Finally, phase five demonstrated the workflow diagram of eLSTM using the proposed MSECosine Loss function. These phases are introduced as below.

A. DATA COLLECTION AND PREPARATION

This section gives the comprehensive details on the descriptions of two collected datasets that used to test the models of this research and also, denote the pre-processing schemes that employed these data to prepare them for prediction purpose.

1) DATASET DESCRIPTION

The research utilizes two time series datasets namely (i) Assessment, and (ii) LMS that are collected by the Infocomm Development Centre (IDEC), Universiti Putra Malaysia (UPM). The description of these dataset is stated in Table 4.

TABLE 4. Time series datasets descriptions.

Name of Dataset	Assessment Dataset	LMS Dataset
Authorship	Infocomm Development Centre (IDEC), Universiti Putra Malaysia (UPM).	Infocomm Developmen t Centre (IDEC), Universiti Putra Malaysia (UPM)
Dataset Characteristics	Multivariate	Multivariate
Attribute Characteristics	Categorical Data (Nominal), Numerical & Continuous Data	Categorical Data (Nominal), Numerical & Continuous Data
Number of Instances	4819	11895
Attributes Number	24	7
Missing Values	No	Yes

The 'Assessment' dataset comprised of 4819 assessment records from 40 courses within 3 semesters in a undergraduate program at the Faculty of Computer Science and Information Technology (FSKTM) at UPM, and consists of 24 attributes of "Faculty name", "Gender", "Age", "type of sponsor", "Assessment of w1-7", "Assessment of w8-12", "CGPA", "Matric" and etc.

The second dataset comprises the access log to the UPM LMS called PutraBLAST. This dataset consists of 11895 instances and contains 7 attributes namely "Time", "Event_context", "Component", "Event_name", "Description", "Origin", and "Matric". Based on the features of this dataset we calculate the frequency of access by each student to generate two features called "week 1-7" and "week 8-12" in each of their registered courses. This information is valuable to indicate the effort of the students in learning as they access the LMS for learning activities.

Once these datasets are combined by mapping them according to the matric number, the total number of instances becomes 3721, and we only utilized 21 attributes that provided more crucial information e.g., the student performance and engagements. The details of the aggregated 'Assessment' and 'LMS' dataset is shown in Table 5.

TABLE 5. Combined datasets descriptions.

Name of Dataset	Combined Assessment & LMS Dataset
Dataset Characteristics	Multivariate
Attribute Characteristics	Categorical Data (Nominal), Numerical & Continuous Data
Number of Instances	3721
Attributes Number	21
Missing Values	No

B. PRE-PROCESSING STEPS

The pre-processing step of this study involved two techniques called LabelEncoder and MinMaxScaler normalization. These methods are beneficial for preparing data for deep time series models. As the collected dataset contains multiple variables, it is necessary to convert categorical features to numerical form to be processed by time series models. Additionally, as time series models are highly sensitive to input scale, the MinMaxScaler normalization method was used to ensure that all features have the same range. A detailed description of each approach is provided in their respective sections.

1) LABEL ENCODER TECHNIQUE

Label Encoder is a technique used to convert categorical data into numerical form. This process involves assigning a unique numerical value to each category present in the dataset used in this research. These numerical values are arbitrary and have no intrinsic meaning. The procedure for converting categorical data into numeric form using LabelEncoder is outlined below.

2) MINMAXSCALAR NORMALIZATION METHOD

MinMaxScaler is a widely used data preprocessing technique in machine learning and data analysis. Its primary objective is to transform the data in a way that all features lie within a specified range.

In our case, we have applied the MinMaxScaler to the combined 'Assessment' and 'LMS' dataset, which contains both categorical and numerical attributes.

The MinMaxScaler method serves to standardize the data, making it amenable for various machine learning algorithms and ensuring that no single feature dominates the learning process due to differences in scale.

Algorithm 1 Steps of Encoding the Features from Categorical form to Numeric

```
Begin
    1. Identify the categorical attributes in the combined dataset 'Assessment' and 'LMS'.
    2. Defined an instance of the LabelEncoder class to use for fitting and transforming the categorical attributes.
    3. Fit the LabelEncoder instance to the categorical column to create a mapping of categories to numerical values.
    4. Transform the categorical column using the fitted LabelEncoder instance to create the numerical column.
    5. Replace the original categorical column with the new numerical column in the dataset.
End
```

This technique operates through three essential steps of 1) Compute minimum (I_{Min}) and maximum (I_{Max}) of input values. These values are critical as they determine the scaling range, 2) Scaling the input data using Eqn. (1) that mentioned below. This equation standardizes each data point in the range [−1, 1] based on its relationship with the minimum and maximum values, 3) Specifying scaling range where in this research, we have chosen to scale the data to the range [−1, 1].

This choice is based on its advantages, such as faster convergence when dealing with deep time series models, which are applied in this work, and the preservation of critical information in the data. The MinMaxScaler transformation can be expressed using the following mathematical formula, as stated in Eqn. (1):

$$I_{Scaled} = \frac{(I - I_{Min})}{(I_{Max} - I_{Min})} \tag{1}$$

where the I represents the input data, I_{Scaled} denotes the scaled data after the transformation, I_{Min} and I_{Max} data signifies the lowest and highest value in the input data respectively.

Therefore, this formula ensures that each feature is scaled proportionally to its range within the dataset, making it a valuable preprocessing step for prediction tasks when using deep time series models.

3) GENERATING STUDENT ENGAGEMENT FEATURES

As explained in the previous sections, the extracted and integrated data are transformed and normalized. We also generated several new features based on the LMS dataset, to represent the learning efforts by the students.

We identified the frequency of weekly access by the students in each course from week 1 until week 12, and created two aggregated features based on their total frequency of access from week 1 until week 7 (called FreqW1W7), and from week 8 until week 12 (called FreqW8W12).

We extracted the first test, continuous assessment and the final grade from the Assessment dataset. The cleaned dataset consists of 21 attributes with the following details:

a) Demographic: Gender, Age, Country, Sponsorship Type, Course Name
b) Engagement: FreqW1, FreqW2, FreqW3, FreqW4, FreqW5, FreqW6, FreqW7, FreqW8, FreqW9, FreqW10, FreqW11, FreqW12, FreqW1W7, FreqW8W12
c) Performance: MarksTest1, MarksContinuous, Grade

4) FEATURE SET GENERATION

To determine the impact of combination of different attributes to the grade prediction model's performance, we designed 29 feature sets shown in Figure. 31. This is also to diminish the risk of overfitting that may occur with a single model. Figure.31 is shown at Appendix.

Figure. 3 illustrates the comprehensive framework employed for student grade prediction, constructed based on five stages: 1) Data collection, 2) Pre-Processing stage, 3) Deep time series models, 4) Prediction stage, and 5) Output. In the initial stage of data collection, the dataset used in this investigation originates from two primary sources, 'Assessment' and 'LMS,' represented as separate time series datasets in the early phase (Refer to Table 4).

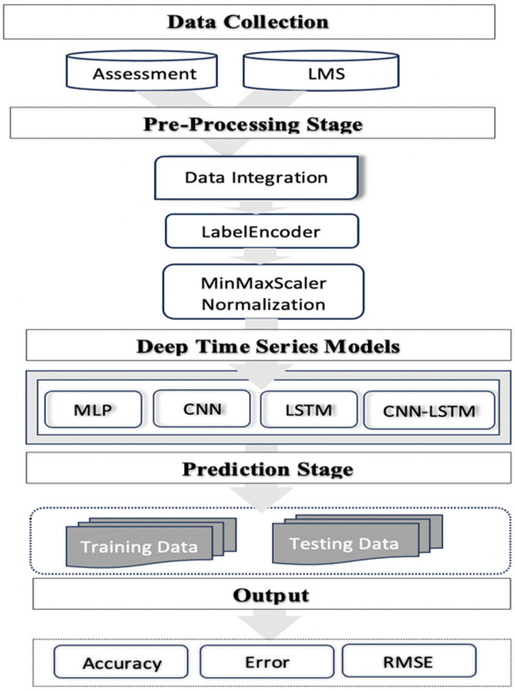


FIGURE 3. The principle and framework of student grade prediction.

Following this, the second stage is the preprocessing stage. In this stage, three traditional preprocessing models of 1) Data Integration, 2) Label Encoding, and 3) Normalization have been employed. In the data integration stage, the aforementioned 'Assessment' and 'LMS' datasets are combined (Assessment & LMS) to provide information about student performance and engagement. Subsequently, distinct techniques of Label Encoding and Normalization are applied to the combined dataset of Assessment & LMS, respectively. The Label Encoding method transforms categorical features into a numerical format compatible with the deep time series models used in this study. Additionally, the MinMaxScaler normalization method is applied to enhance model performance and convergence. This normalization technique is chosen for its suitability in scaling input data to a specific range, recognizing the impact of input scale on deep time series models.

Moving forward, the third phase involves the development of four deep time series models: MLP, CNN, LSTM, and CNN-LSTM for student grade prediction. These models are systematically split into training and testing sets during the prediction stage, ensuring an unbiased assessment of their effectiveness and guarding against overfitting.

The output stage concludes the research framework, providing accuracy reports based on predictions made on the testing data. Evaluating the performance of each deep time series technique yields insights into the most effective approach. The model demonstrating higher accuracy and lower error values is selected as the optimal scheme for student early-grade prediction. Subsequently, this chosen approach is refined further through the proposed MSECosine loss function, detailed in the subsequent section outlining the procedures taken to enhance the selected model.

5) SETUP OF DEEP TIME SERIES MODLES FOR GRADE PREDICTION

The combined Assessment & LMS dataset is carefully divided into 70% for training our models and 30% for testing to make sure our models are well-trained and thoroughly evaluated. After that, we created four different deep learning models: MLP, CNN, LSTM, and CNN-LSTM. To provide transparency in our model configurations, Table 6, outlining both tested and chosen values for key hyperparameters crucial for optimizing the performance of each time series model.

TABLE 6. Tested hyperparameters for model configuration optimization.

Parameters	Tested values	Chosen values
Sequence Length(timestep)	50-150	100
learning rate	0.0001-0.0003	0.0003
batch size	32,64,128, and 1024	1024
epoch number	10	10
Optimizer	Stochastic Gradient Descent (SGD)	-------

All these time series models are built using python programming languages using various libraries. The "Pandas" library due to providing several functions to load data in various formats e.g., CSV is utilized to load the time series dataset of this work. Then, for the preprocessing stage the {"NumPy" and "Sklearn"} libraries have been utilized. Furthermore, beside these libraries, another three libraries of {"Keras"," TensorFlow", and "Matplotlib"} were used to build the four deep time series approaches.

The MLP model utilized in this research comprises a single layer with 100-time steps, 21 features based on a designed predictive analysis model of 29 (this may vary depending on the selection of the designed model), and one channel. It is followed by one fully connected layer with 79 units and single output layer with a single unit.

The CNN model consists of 1D convolutional layer with 64 filters, a kernel size of 2, and a LeakyReLU activation function is used, subsequently one max pooling layer with a pool size of 2, one flattened layer, and one output layer with a single unit.

Similarly, the LSTM model comprises one LSTM layer comprising 79 units and a LeakyReLU activation function, afterwards one flattened layer and one output layer with a single unit.

Also, the CNN-LSTM technique contains a single distributed convolutional layer containing 64 filters, each with a kernel size of 1, and a LeakyReLU activation function, followed by a singular distributed max pooling layer with a pool size of 2, a singular distributed flatten layer, one LSTM layer with 79 units and LeakyReLU activation function, one flattens layer, and one output layer with a single unit.

6) EVALUATION METRICS

The performance of the used deep time series models is evaluated using accuracy (Eqn. (2)), MSE (Eqn. (3)) and RMSE (Eqn. (4)) as follows.

$$Accuracy = \frac{1}{T} x \sum [g_{pred\,l} == g_l] \tag{2}$$

where T is the size of testing set, $[g_{pred_l} == g_l]$ is an indicator function. If the indicator function is equal to 1, it signifies that the predicted grade g_{pred_l} for students l is equivalent to the true grade g_l. Conversely, if the indicator function is equal to 0, it indicates a mismatch between the predicted and true grades. Finally, the term $\frac{1}{T}$ represents the computation of the average accuracy by dividing the sum of indicator functions by the total number of instances (T).

Eqn. (3) represents the MSE formula, which quantifies the error between the true and predicted grades.

$$MSE = \frac{1}{T} x \sum [g_l - g\_pred_l]^2 \tag{3}$$

where T is specified, the testing set size and Σ denotes the sum over all student's grades in the testing set. Also, g_l and g_{pred_l} are denoting the true and predicted grade of student l, respectively. The MSE is calculated as the average of the squared differences between the true and predicted grades.

Eqn. (4) shows the RMSE formula which calculates the square root of the average squared differences between the predicted g_pred_l and actual g_l values.

$$RMSE = Sqrt\left(\left(\frac{1}{T}\right) x \sum |g_l - g\_pred_l|^2\right) \tag{4}$$

where T is the observations number, g_pred_l and g_l are represent the predicted and actual values respectively.

The deep time series model that has the best performance is then selected to be improved with proposed MSECosine loss function as explained in the next section.

IV. ENHANCEMENT OF LSTM WITH MSECOSINE LOSS FUNCTION

A. PROPOSED MSECOSINE LOSS FUNCTION

In implementing the eLSTM model, we utilized several Python libraries, including Pandas, NumPy, Scikit-learn, Keras, TensorFlow, and Matplotlib. These libraries were employed for efficiency and adhering to standard practices in data processing and model construction. Notably, the core mathematical components of Algorithm2, particularly the proposed MSECosine loss function, considered as a custom loss function, were implemented from scratch.

Referring to Eqn. (3), if the error obtained by the MSE loss function has large value, then the MSE loss function due to having the square term, will amplify the error even further. This will cause numerical instability and slow convergence. Hence, in order to address this concern, the present study introduced a novel MSECosine loss function.

The proposed MSECosine loss function of this study is constructed by combination of two loss functions of MSE and LogCosh. This combination has been done based on the convexity theorem where this theorem allows the convex functions to be combined together in a specific way while preserving their desirable properties and convexity nature.

Therefore, the proposed MSECosine loss function can offer the ability of controlling the errors from growth and prevent the function from being too sensitive to the outliers. The proposed MSECosine function gets these advantages from the MSE and LogCosh loss function respectively. So, it can be more robust and be able to handle a wide range of data rather than using the MSE and LogCosh functions individually.

The below steps present the mathematical proof of the convexity of the proposed MSECosine function based on the convexity theorem. Let the proposed MSECosine defined as $MSECosine\left(g_l - g_{pred_l}\right)$, where the g_l and g_{pred_l} be the actual and predicted value correspondingly.

$$\begin{aligned} & MSECosine\left(g_l - g_{pred_l}\right) \\ & \quad = MSE\left(g_l, g_{pred_l}\right) + (1- \propto)\,\mathrm{x}LogCosh\left(g_l, g_{pred_l}\right) \end{aligned} \tag{5}$$

where the $\propto$ is a hypermeter which is between 0 to 1. This value is controlling the relative weights of the proposed MSECosine function. While the $\propto$ is close to 0 the MSECosine function gets the advantage from the LogCosh function to deal with any outlier. Otherwise, if the alpha value is close to one, they get benefits from the MSE function. g_l and g_{pred_l} are signifying the actual and predicted value over 100-time steps of this study.

Thus, to proof that $MSECosine\left(g_l - g_{pred_l}\right)$ is convex, the Hessian matrix must be positive semidefinite for each value of g_l and g_{pred_l}. The Hessian matrix of $MSECosine\left(g_l - g_{pred_l}\right)$ with regards to g_{pred_l} is specified as follows.

$$\begin{aligned} h &= \partial^2 \frac{MSECosine\left(g_l - g_{pred_l}\right)}{\left(\partial g_{pred_l}\right)^2} \\ &= \frac{\partial^2\left[MSE\left(g_l - g_{pred_l}\right) + (1- \propto)\,\mathrm{x}LogCosh\left(g_l - g_{pred_l}\right)\right]}{\left(\partial y_{pred_T}\right)^2} \\ &= \frac{\partial^2 MSE\left(g_l - g_{pred_l}\right)}{\left(\partial g_{pred_l}\right)^2} + \frac{(1- \propto)\,\mathrm{x}LogCosh\left(g_l - g_{pred_l}\right)}{\left(\partial g_{pred_l}\right)^2} \\ &= \frac{2\mathrm{x}M + \frac{\propto \mathrm{x} sin^2\left(\left(g_l - g_{pred_l}\right)\right)}{\propto}}{Cos^3\left(\left(\frac{g_l - g_{pred_l}}{\propto}\right)\right)} \end{aligned} \tag{6}$$

Here the M is meant for matrix. Subsequently, as the both MSE and LogCosh as well as all the values of $g_l - g_{pred_l}$ are positive semidefinite in h, so the proposed $MSECosine$ loss function is convex.

B. ENHANCED LSTM MODEL

In this study, four deep time series models namely, MLP, CNN, LSTM, and CNN-LSTM have been employed for early student grade prediction using the combined 'Assessment' and 'LMS' dataset. Then, the selected method of this study with higher precision which is LSTM is getting enhanced by the proposed loss function of MSECosine that is constructed by combination of two popular loss functions of MSE and LogCosh. This step was accomplished for testing which can offer more precise prediction while applying the proposed MSECosine loss function to the standard LSTM technique.

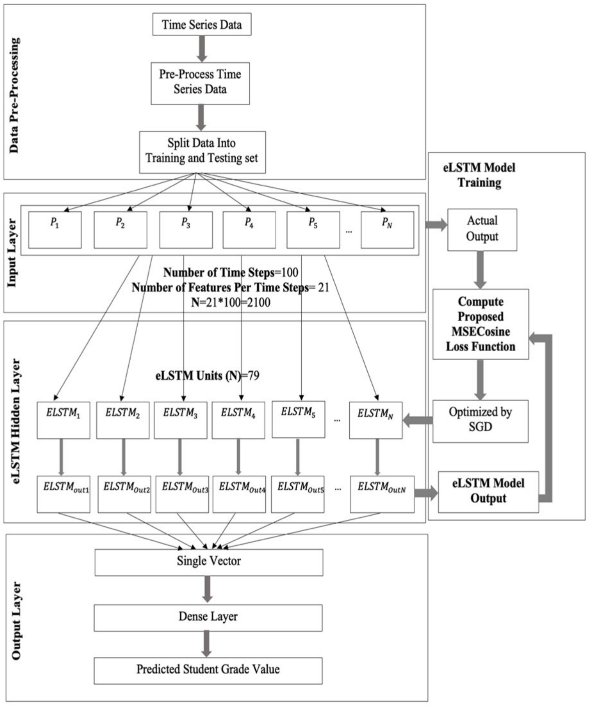


FIGURE 4. eLSTM model using proposed MSECosine loss function.

Figure. 4 illustrates the architecture diagram of the eLSTM scheme.

According to Figure. 4, the eLSTM model of this study consists of five stages of 1) Pre-processing, 2) Input layer, 3) eLSTM hidden layer, 4) Output layer, and 5) eLSTM Model Training based on 70% and 30% train-test split.

The pre-processed data that is fed to the input layer stage is set to the input value of 100 data points and the number features is obtained from the designed feature selection models (Refer to Figure. 31). Also, to obtain the best time step value, various timestep values were tested and value of 100 is considered as the best time step value.

In this layer, the input shape is calculated based on the multiplication of time steps, which is set to 100, with varying feature values ranging from 5 to 21 per time step. These feature values are obtained from the designed models (Refer to Figure. 31). Therefore, the total number of features steps per time step is 2100.

Afterward, the third stage belonged to the eLSTM model hidden layer where in here we only have one hidden layer as it most of study claimed that the LSTM with a single hidden layer provided more accurate results [1]. This layer has 79 LSTM units that take the varying feature values ranging from 5 to 21 features $(P_1, P_2, \ldots, P_{21})$ from the designed feature selection model of 29 as input for each 100-time steps. Afterwards, the eLSTM units using their memory cells process the input sequence to update the hidden states and yield a new hidden state and a new memory cell for each 100-time steps. Subsequently, the output from the last time step of each eLSTM unit is associated with the next layer of output layer.

In the output layer, the obtained output from the last time step of each eLSTM unit is concatenated into a single vector. This vector is then passed over a dense layer with one output neuron. The output of this neuron shows the predicted student grade value of the target variable which is the grade.

Finally, in the training stage, the Stochastic Gradient Descent (SGD) [1] has been used as an optimizer and the MSE for the loss function [18], [34]. In this stage the main concern is to make the predicted student grade as close as possible to the actual grade by adjusting the model's weights. This can be done by computing the loss function and updating the weights using backpropagation.

The details of the proposed eLSTM model based on the 29 designed feature selection models of this study is explained in Algorithm 2.

The proposed eLSTM model is get a sequence of observations $P = \{P_1, P_2, P_3, \ldots, P_N\}$ as an input, where each of these observations are a vector with k length. Then, the v_0 and j_0 present the initial hidden state and cell state respectively. Each of these two parameters are a vector with length of s.

The mentioned input and hidden state are altered by usage of various weights metrics of we_f, we_i, we_c, and we_o with size of $sx(k+s)$, and we_y that has size of $1xn$. Also, there are multiple bias vectors ba_f, ba_i, ba_c, and ba_o with size of s, and which has size 1.

Hence, the variety of these weights and biases are because they must learn different transformations for the defined input and hidden state to attain an appropriate activation of gates and output in each time step.

Formerly, defined the number of time steps in which in this work is equal to 100. Furthermore, map the hidden state to the prediction. By tuning these parameters during the training stage, the LSTM model can yield more precise predictions based on the input sequence of P.

Consequently, the order of forecasted outcomes $y = \{y_1, y_2, y_3, \ldots, y_N\}$ where each of these predicted values are a scalar in which they are considered as the output.

Once the predicted value obtains from the eLSTM model after 100-time steps then the eLSTM model employed the proposed MSECosine loss function to compute the differences among the predicted value of y_pred_T and true value y_T.

Algorithm 2 Proposed eLSTM Model Steps

```
for $t$ in range $(1, t+1)$:
    #Concatenate the input and prior hidden state into a new vector,
        $Q_T = \text{np.concatenate}((P_T, v_T - 1))$
    #Calculate input gate activation that used sigmoid function $\sigma$,
        $f_T = sigmoid(np.dot(we_f, Q_T) + ba_f)$
    #Calculate input gate activation by using sigmoid function $\sigma$,
        $i_T = sigmoid(np.dot(we_{i,} Q_T) + ba_i)$
    #Calculate candidate cell state $\pi c_T$ using tanh function
        $\pi c_T = np.\text{tanh}(\text{np.dot}(we_c, Q_T) + ba_c)$
  #Adjust cell state at time step $c_T$ using element-wise
  #multiplication symbol $\odot$,
        $c_T = f_T \bigodot c\,(T-1) + i_T \odot \pi c_T$
    #Calculate the output gate activation $O_T$ using sigmoid
    #function $\sigma$,
        $O_T = sigmoid(np.dot(we_O, Q_T) + ba_O)$
    #Tune the hidden state $v_T$ using element wise multiplication
    #$\odot$ combined with the hyperbolic tangent function,
      $v_T = O_T * \text{np.tanh}(c_T)$
    #Compute prediction
      $g\_pred_T = np.dot(we_y, Q_T) + b_{ay}$
    #Compute the proposed MSECosine loss function $L_T$
    #between the predicted and actual label value
      $L_T = MSECosine(g\_pred_T, Actual\_lable_T)$
    #Iterate through all previous steps for each 100-time
    #steps $T = 1, 2, 3, .., t$
    #Return the sequence of predictions $g\_pred_T$.
```

Therefore, the predicted sequence is describe as $g\_pred_T = \{g1_{pred\,T}, g2_{pred\,T}, \ldots, gT\_pred_t\}$ and the target sequence $g_T = \{g1_T, g2_T, g3_T, \ldots, g_t\}$, then the proposed MSECosine loss function is specified in Eqn. (7),

$$\begin{aligned} MSECosine\left(g_T - g_{pred_T}\right) &= MSE\left(g_T, g_{pred_T}\right) \\ &\quad + (1 - \propto)\,\text{x}LogCosh\left(g_T, g_{pred_T}\right) \end{aligned} \tag{7}$$

where Σ is sum over all 100-time steps of this study $T = 1, 2, 3, .., t$.

V. EXPERIMENT AND RESULTS

This section provides the setting and outcomes of three experiments conducted in this study which are the evaluation of deep time series approaches on 29 designed feature set models, comparison of the eLSTM method against the basic deep time series approaches, and the performance of the models on the feature sets.

A. EXPERIMENT 1: ENHANCED LSTM MODEL EXPLANATION

This section outlines the process to identify the optimal deep time series model among MLP, CNN, LSTM, and CNN-LSTM. To evaluate these models, 29 distinct feature selection schemes were devised, utilizing a combined dataset ('Assessment' and 'LMS') (Figure. 31).

Data normality was assessed using box plots, providing MIN, Quartile1, Median, Quartile3, MAX, and Mean values. Comparison of mean and median values indicates normality; disparities prompt non-parametric tests.

Table 7, 8, and 9 display minimum, quartile1, median, quartile3, maximum, and mean values from experimental outcomes for MLP, CNN, LSTM, and CNN-LSTM across 29 feature sets. Corresponding box plots are illustrated in Figures 5, 6, and 7 (based on Table 7, 8, and 9).

TABLE 7. Descriptive statistic of accuracy based on 29 features sets.

Model	MLP	CNN	LSTM	CNN-LSTM
MIN	0.5571	0.5055	**0.6022**	0.6013
Q1	0.5856	0.5994	**0.6123**	0.6068
Med	0.6049	0.6041	**0.6142**	0.6100
Q3	0.6087	0.6087	**0.6169**	0.6151
Max	0.6133	0.6133	**0.6225**	0.6179
Mean	0.5954	0.5984	**0.6134**	0.6106

TABLE 8. Descriptive statistic of error based on 29 features sets.

Model	MLP	CNN	LSTM	CNN-LSTM
MIN	0.2362	0.2345	**0.2335**	0.2345
Q1	0.2393	0.2394	**0.2345**	0.2356
Med	0.2427	0.2425	**0.2356**	0.2367
Q3	0.2464	0.2462	**0.2374**	0.2385
Max	0.2772	0.3097	**0.2335**	0.2431
Mean	0.2442	0.2464	**0.2374**	0.2376

TABLE 9. Descriptive statistic of RMSE based on 29 features sets.

Model	MLP	CNN	LSTM	CNN-LSTM
MIN	0.4753	0.4920	**0.4839**	0.4919
Q1	0.4917	0.4968	**0.4851**	0.4961
Med	0.4937	0.5009	**0.4858**	0.5009
Q3	0.5022	0.5105	**0.4872**	0.5105
Max	0.5972	0.5782	**0.4929**	0.5782
Mean	0.5040	0.5089	**0.4867**	0.5088

Referring Tables of 7, 8, and 9, and Figures. 5, 6, and 7, we can conclude that the LSTM model produced higher performance compared to the three tested models by offering higher accuracy, Lower error and RMSE values. Therefore, we selected the LSTM model as the best model for early student grade prediction.

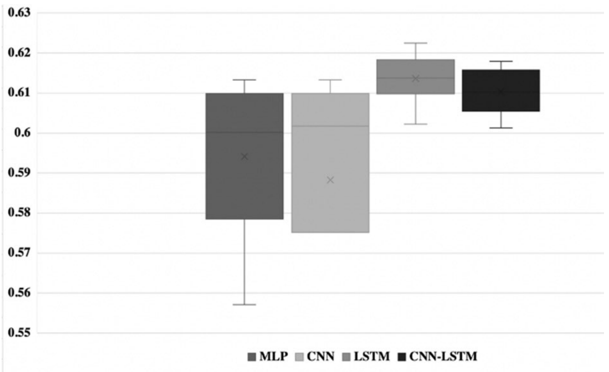


FIGURE 5. Box plot of accuracy using 29 feature sets.

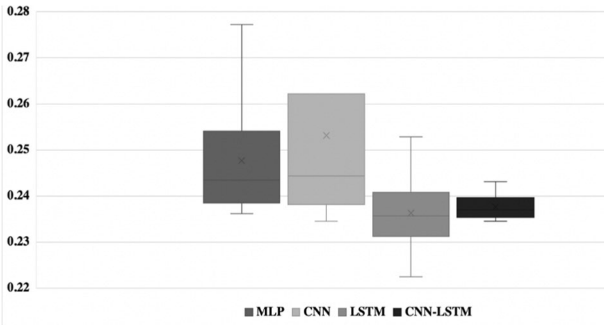


FIGURE 6. Box plot of error using 29 feature sets.

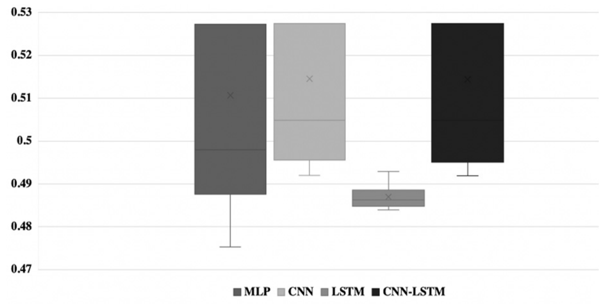


FIGURE 7. Box plot of RMSE using 29 feature sets.

B. EXPERIMENT 2: PERFORMANCE OF eLSTM MODEL COMPARED TO STANDARD LSTM MODEL

This section specified the comparison results between the standard LSTM and proposed eLSTM model based on three terms of accuracy, MSE, and RMSE. Two sub-experiments were conducted as follow:

- Experiment 2.1: LSTM vs eLSTM using all 29 feature sets
- Experiment 2.2: Effect of Proposed Loss function

1) EXPERIMENT 2.1: EFFECT OF DEMOGRAPHY LSTM VS ELSTM USING FEATURE SET 29

Table 10 present the comparison of accuracy, Error, and RMSE of LSTM to the proposed eLSTM model based on six values of MIN, Quartile1, Median, Quartile3, MAX, and Mean. Figures. 8, 9, and 10 demonstrate the box plot of the standard LSTM and eLSTM models based on Table 10 respectively.

TABLE 10. Comparison of accuracy, MSE, RMSE based on 29 features sets.

	Accuracy		MSE		RMSE	
Model	LSTM	eLSTM	LSTM	eLSTM	LSTM	eLSTM
MIN	0.6022	0.6123	0.2335	0.1724	0.4839	0.4835
Q1	0.6123	0.6169	0.2345	0.1730	0.4851	0.4844
Med	0.6142	0.6184	0.2356	0.1736	0.4858	0.4861
Q3	0.6169	0.6215	0.2374	0.1746	0.4872	0.4871
Max	0.6225	0.6271	0.2335	0.1764	0.4929	0.4935
Mean	0.6134	**0.6191**	0.2499	**0.1738**	0.4867	**0.4862**

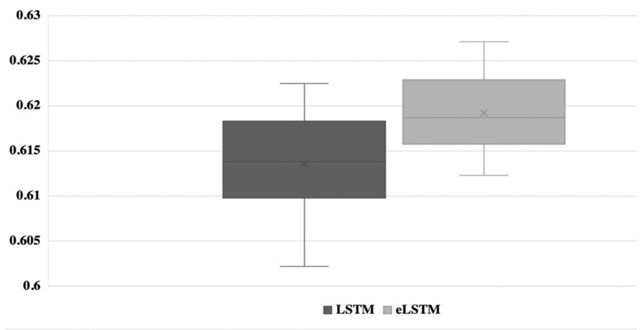


FIGURE 8. Box plot of accuracy between LSTM and eLSTM using 29 feature sets.

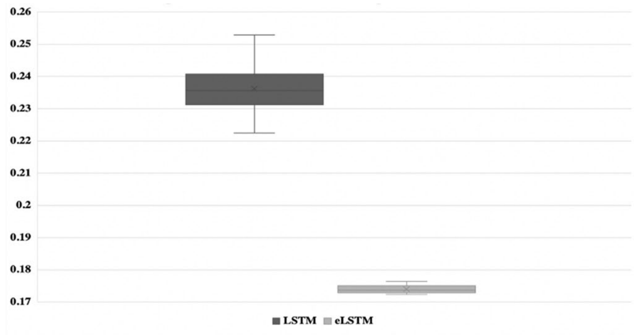


FIGURE 9. Box plot of error between LSTM and eLSTM using 29 feature sets.

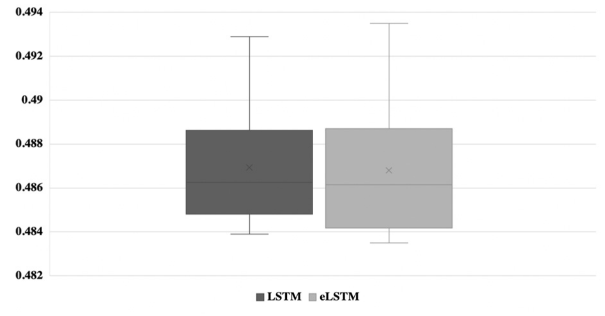


FIGURE 10. Box plot of RMSE between LSTM and eLSTM using all 29 feature sets.

Pointing out Figures. 8,9, and 10, It can be inferred that the introduced eLSTM technique exhibited superior performance in comparison to the standard LSTM model in terms of higher accuracy, and lower error and RMSE values.

Deriving insights from the Tables of 7 to 10, and Figs. 5-11, the mean and median values of each time series technique are unequal. Accordingly, to assess that the MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM methods are dissimilar, the Friedman test which considers as the most well-known approach for testing the dissimilarity among more than one sample utilization. Table 11 indicates the mean of rank using the Friedman test with the p-value for MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM accuracy based on 29 designed feature selection methods.

TABLE 11. Mean of rank by Friedman test with the p-value for MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM Models based on all 29 designed feature selection methods.

Sample	Frequency	Sum of ranks	Mean of ranks
eLSTM	29	29.000	1.000
LSTM	29	58.000	2.000
CNN-LSTM	29	87.000	3.000
CNN	29	116.000	4.000
MLP	29	145.000	5.000
p-value (one-tailed)	0.0043		
alpha	0.05		

Referring to Table 11, this evaluation confirms that the time series models (MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM) are significantly different. The p-values of their precisions are resulting in the null hypothesis being rejected due to an alpha value below 0.05. Therefore, we proceed with five post hoc tests (Nemenyi, Bonferroni–Dunn, Finner, Li, and Holm) to obtain specific pairwise comparisons and identify the observed differences.

Tables 12 and 13 present the adjusted p-values to test multiple comparisons among five deep time series models of MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM. The proposed eLSTM model due to having the smallest mean of rank compared with the other tested techniques of MLP, CNN, LSTM, and CNN-LSTM is taken as the control method. Tables 12 and 13 demonstrates the highly significant improvement of the proposed eLSTM method over MLP, CNN, LSTM, and CNN-LSTM with the significant level $\alpha = 0.05$.

TABLE 12. Adjusted p-value for tests for multiple comparisons based on all 29 feature sets.

Model	MLP	CNN	LSTM	CNN-LSTM	eLSTM
MLP	1.0000	0.9000	0.0217	0.1621	0.0010
CNN	0.9000	1.0000	0.0290	0.1979	0.0010
LSTM	0.0217	0.0290	1.0000	0.9000	0.2111
CNN-LSTM	0.1621	0.1979	0.9000	1.0000	0.0319
eLSTM	0.0010	0.0010	0.2111	0.0318	1.0000

Referring to obtained results from the Tables mentioned above, the proposed eLSTM model presented the

TABLE 13. Adjusted p-value for tests for multiple comparisons among four models of MLP, CNN, CNN-LSTM, and proposed eLSTM based on all 29 feature selection methods.

Methods	Bonferroni Dunn Test	Finner Test	Li Test	Holm Test
	Adjusted P-value			
eLSTM vs MLP	1.346486e-07	1.346485e-07	2.284724e-08	1.346486e-07
eLSTM vs CNN	2.267562e-07	1.346485e-07	3.847611e-08	1.889635e-07
eLSTM vs LSTM	1.065728e-01	1.776214e-02	1.966749e-03	1.776214e-02
eLSTM vs CNN-LSTM	1.161373e-02	3.867497e-03	1.776214e-02	7.742487e-03

improvement of 3.97632%, 3.45922%, 0.929247%, and 0.503247% in terms of accuracy in comparison to other techniques of MLP, CNN, LSTM, and CNN-LSTM respectively (Referring to Table 7).

The details of the results of each of these stated methods are represented in Appendix. Therefore, based on the provided results in appendix, the designed feature selection 11 (refer to Figure. 31) produced higher accuracy in both models of LSTM and proposed eLSTM. Thus, we can conclude that the student engagement in the first week has a major impact on their overall performance.

2) EXPERIMENT 2.2: EFFECT OF PROPOSED LOSS FUNCTION

To show the superiority of the proposed MSECosine loss function compared to state of art loss functions in terms of addressing the vanishing gradient issue, we monitor the loss trend value. Figure.11 demonstrated the comparison of the various loss of MSE and MAE loss functions with the proposed MSECosine loss function of this work. We used feature set 11 since it has the best accuracy to further investigate the effectiveness of the proposed MSECosine against the standard approach.

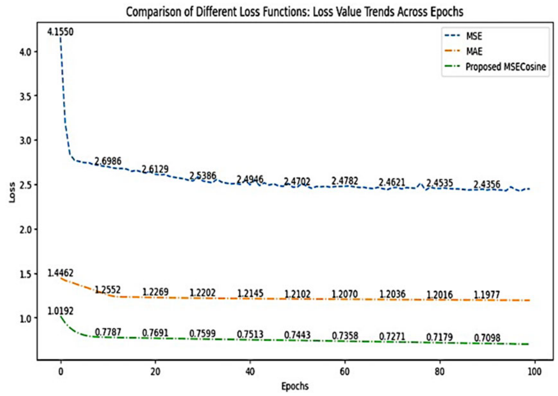


FIGURE 11. Comparison of loss trends value across each epoch between MSE, MAE, and proposed MSECosine loss functions.

Referring to Figure. 11, all three loss functions of MSE, MAE, and the proposed MSECosine demonstrated a decrease over 100 epochs. However, the initial values differed, with MSE starting at 4.1550, MAE at 1.4462, and the proposed MSECosine at 1.0192. At the end of the training, the final values for MSE, MAE, and the proposed MSECosine were 2.4356, 1.1977, and 0.7098, respectively. This analysis underscores the superiority of our proposed MSECosine loss function in effectively mitigating the vanishing gradient issue during training.

C. EXPERIMENT 3: EFFECT OF FEATURE SETS ON EARLY GRADE PREDICTION

The combination of feature sets plays a significant role in the performance of the early grade prediction as student's engagement and attainment develops over the semester [8].

Based on the proposed eLSTM model, which focuses on leveraging the model's ability to capture temporal dependencies and patterns, we found out that seven attributes {Gender, Age, Country, Sponsorship Type, Course Name, MarksTest1, FreqW1} are the most important features for the early grade prediction.

This is because the designed Feature Set 11 that consist of these seven mentioned attributes produced highest accuracy and lowest error and RMSE value. Therefore, we conducted further experiments to observe more closely the performance of the features as below:

- Experiment 3.1: Effect of demography
- Experiment 3.2: Effect of demography, weekly engagement in week 1 until 7, Test1 marks
- Experiment 3.3: Effect of demography, weekly engagement in week 1 until 7, Test1 marks and FreqW1W7
- Experiment 3.4: Effect of demography, weekly engagement in week 8 until 12 and continuous marks
- Experiment 3.5: Effect of demography, weekly engagement in week 8 until 12, continuous marks and Freq 8 to 12
- Experiment 3.6: Effect of using all features

The detailed design of the experiments is provided below:

1) EXPERIMENT 3.1: EFFECT OF DEMOGRAPHY

This experiment is based on models developed using data prepared based on Feature Set *1*. Figures. 12,13, and 14 display the comparisons of five models of MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM based on the accuracy, error, and RMSE.

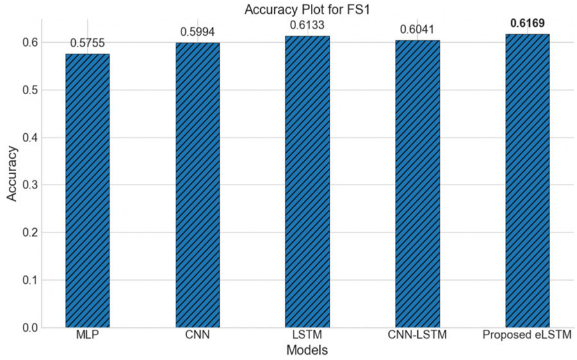


FIGURE 12. Accuracy comparison based on feature set 1.

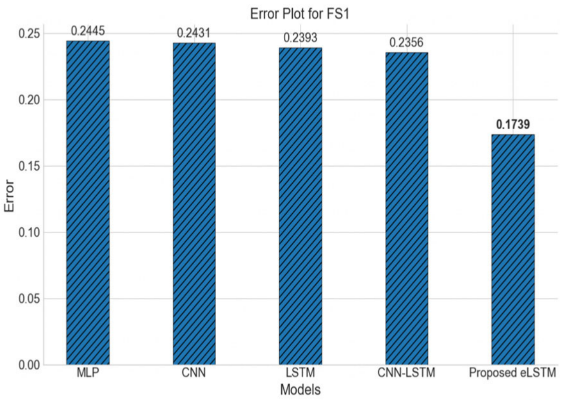


FIGURE 13. Error comparison based on feature set 1.

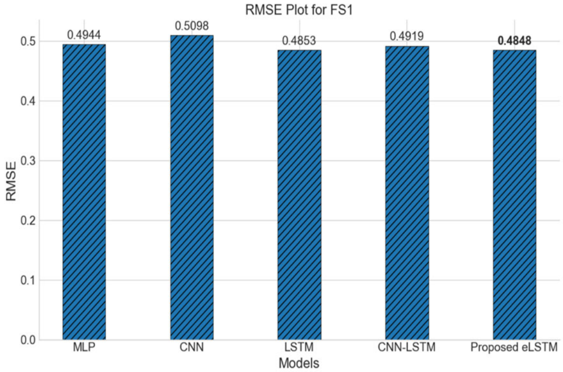


FIGURE 14. RMSE comparison based on Feature Set 1.

Referring to the Figures. 12,13, and 14, the proposed eLSTM model yields higher accuracy and lower error and RMSE value on the designed *Feature Set* 1 compared to other four tested time series model. Therefore, it signifies that the proposed eLSTM technique showed superiority on demographic categories.

2) EXPERIMENT 3.2: EFFECT OF DEMOGRAPHY, WEEKLY ENGAGEMENT IN WEEK 1 UNTIL 7, TEST1 MARKS

This experiment is based on models developed using data prepared based on Feature Set 2 until 9. Similarly, to pervious experiment, Figures. 15, 16, and 17 illustrate the comparisons of five models of MLP, CNN, LSTM, CNN-LSTM, and proposed eLSTM based on the accuracy, error, and RMSE.

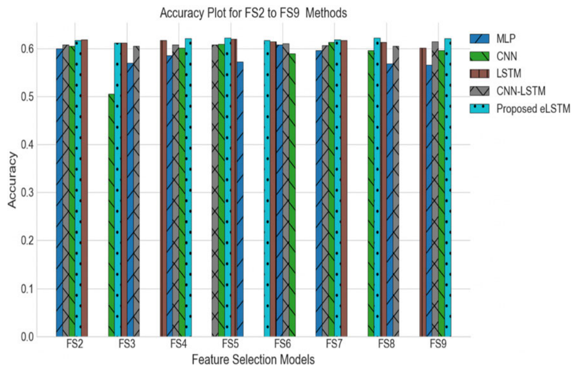


FIGURE 15. Accuracy comparisons based on Feature Set 2 to 9.

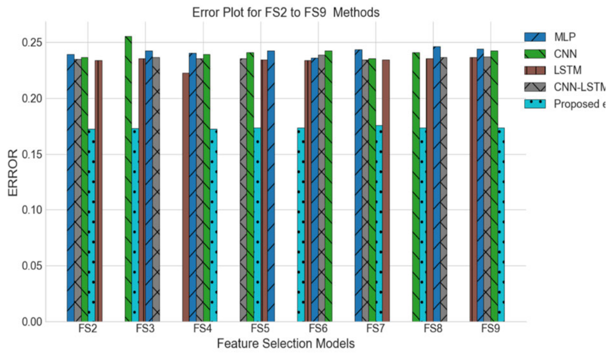


FIGURE 16. Error comparisons based on Feature Set 2 to 9.

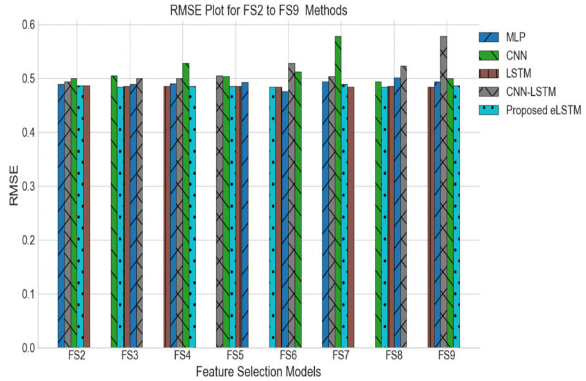


FIGURE 17. RMSE comparisons based on Feature Set 2 to 9.

Regards to the above-mentioned figures, the proposed eLSTM model better performance compared to other four tested models on feature set 4 and 5 in terms of producing higher accuracy.

3) EXPERIMENT 3.3: EFFECT OF DEMOGRAPHY, WEEKLY ENGAGEMENT IN WEEK 1 UNTIL 7, TEST1 MARKS AND FREQW1 TO W7

This experiment is based on models developed using data prepared based on feature set 10 until 17, which extends feature set 2 until 9 with an additional feature that comprises

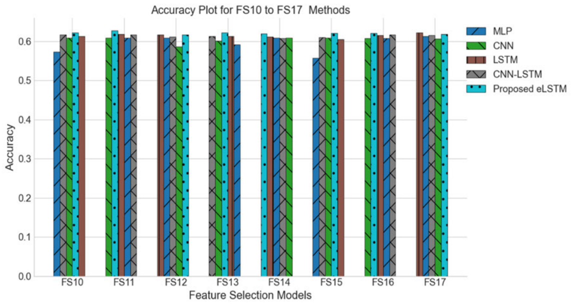


FIGURE 18. Accuracy comparisons based on Feature Set 10 to 17.

of the total frequency of LMS access between week 1 until 7. Figures. 18, 19, and 20 present the comparisons of five models of MLP, CNN, LSTM, CNN-LSTM, and Proposed eLSTM based on the accuracy, error, and RMSE.

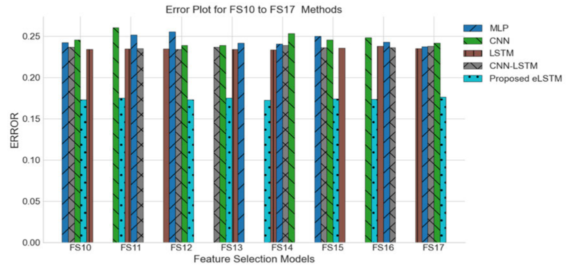


FIGURE 19. Error comparisons based on Feature Set 10 to 17.

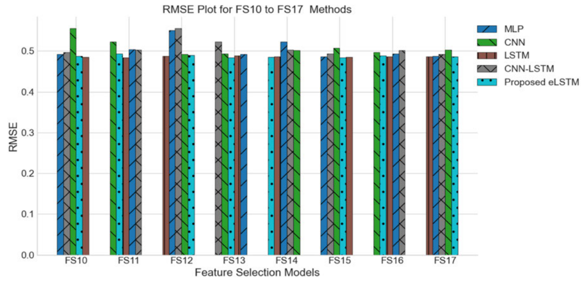


FIGURE 20. RMSE comparisons based on Feature Set 10 to 17.

Based on the above-mentioned figures the proposed eLSTM model showed higher performance on designed feature sets of 11 compared to all other models (in all experiment 1 until 3).

4) EXPERIMENT 3.4: EFFECT OF DEMOGRAPHY, WEEKLY ENGAGEMENT IN WEEK 8 UNTIL 12, AND CONTINOUS MARKS

This experiment is based on models developed using data prepared based on feature set 18 until 23.

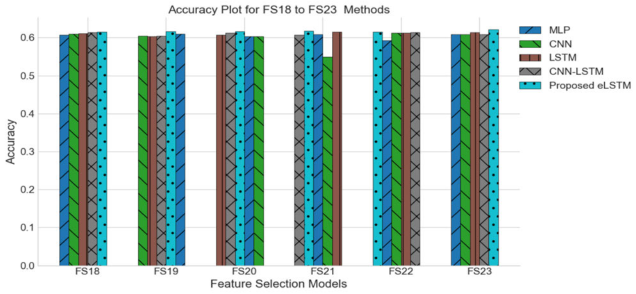


FIGURE 21. Accuracy comparisons based on Feature Set 18 to 23.

Refer to above figures the eLSTM scheme showed higher performance in terms of accuracy on designed feature set model of 23.

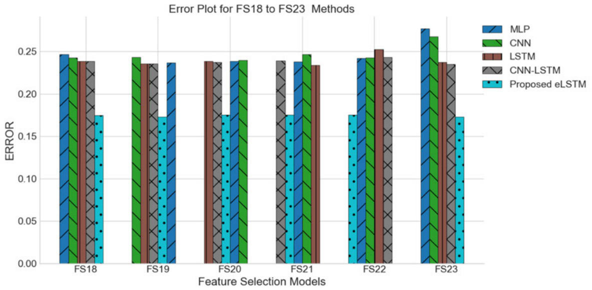


FIGURE 22. Error comparisons based on Feature Set 18 to 23.

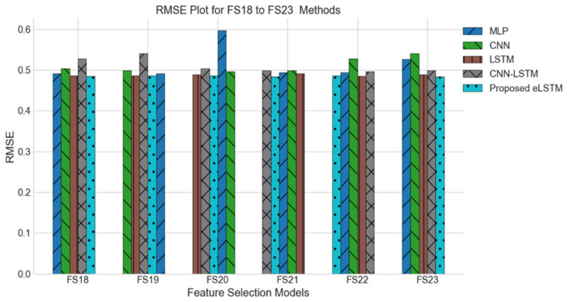


FIGURE 23. RMSE comparisons based on Feature Set 18 to 23.

5) EXPERIMENT 3.5: EFFECT OF DEMOGRAPHY, WEEKLY ENGAGEMENT IN WEEK 8 UNTIL 12, AND CONTINOUS MARKS AND FREQW8 TO W12

This experiment is based on models developed using data prepared based on feature set 24 until 28.

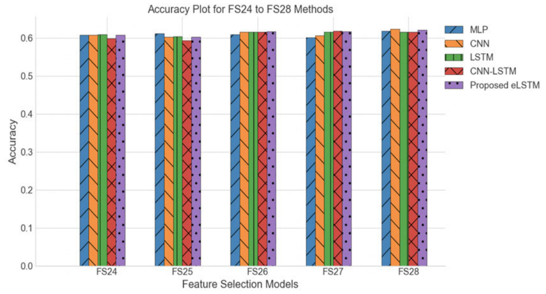


FIGURE 24. Accuracy comparisons based on Feature Set 24 to 28.

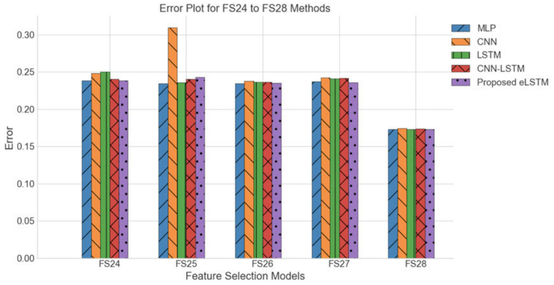


FIGURE 25. Error comparisons based on Feature Set 24 to 28.

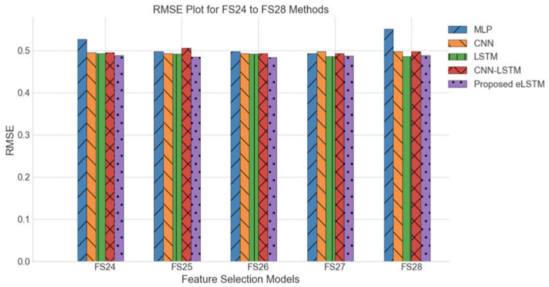


FIGURE 26. RMSE comparisons based on Feature Set 24 to 28.

Refer to Figures 24, 25, and 26, the highest accuracy is belonged to designed feature set of 25 using proposed eLSTM model.

6) EXPERIMENT 3.6: EFFECT OF ALL FEATURES

This experiment is based on models developed using all the attributes in the feature sets.

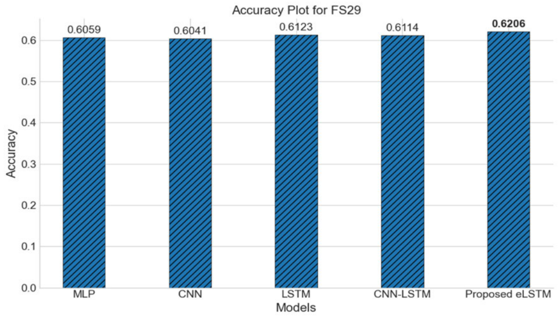


FIGURE 27. Accuracy comparison based on Feature Set 29.

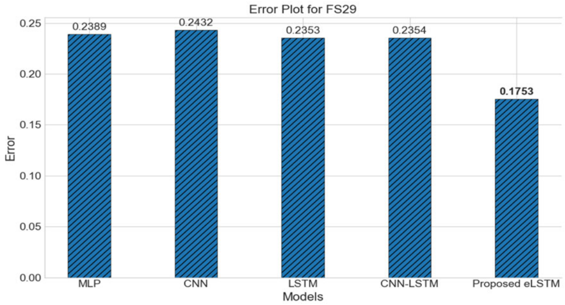


FIGURE 28. Error comparison based on Feature Set 29.

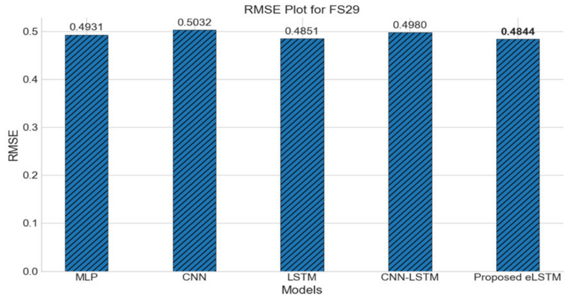


FIGURE 29. RMSE comparison based on Feature Set 29.

Referring to the above stated figures the proposed eLSTM model produced higher performance compared to other approaches of MLP, CNN-LSTM, and CNN-LSTM in terms of accuracy, error, and RMSE value on designed feature model of 29.

VI. DISCUSSIONS

The LSTM model outperforms MLP, CNN, and CNN-LSTM, achieving superior accuracy with reduced errors and RMSE. This is further enhanced by the proposed MSECosine loss function. Across six experiments with varied feature sets, the eLSTM model consistently excels in accuracy, error, and RMSE compared to MLP, CNN-LSTM, and CNN-LSTM.

Feature set 11, combining demography, weekly engagement, as FreqW1W7 and Test1 marks, yields the highest accuracy. To gain insights into eLSTM's performance and feature influence, we employ the LIME library, highlighting the significance of Feature Set 11. The eLSTM's workings, driven by this feature set, illustrate its effectiveness, supported by LIME's interpretability which is mentioned in Appendix.

To establish the superiority of the proposed MSECosine loss function in addressing the vanishing gradient issue compared to state-of-the-art loss functions, we monitored the trend values of the loss. Figure 11 presents a comparative analysis of the loss trends between MSE, MAE loss functions, and the proposed MSECosine loss function in this study. Feature Set 11, known for its highest accuracy, was specifically chosen to delve deeper into the effectiveness of the MSECosine approach compared to the conventional method.

The proposed eLSTM technique in this study employs Lime explanation, comprising three key elements: 1) Instance explanation, 2) LIME explanation, and 3) Prediction Probability. In the "Instance explanation," eight features—Gender, Age, Country, Sponsorship Type, Course Name, MarksTest1, FreqW1, and FreqW1W7—are presented to elucidate the model's prediction. LIME generates explanations by altering feature values and observing prediction changes.

Within the instance explanation, features are paired, each assigned an importance weight indicating its impact on prediction. Positive weights indicate higher predicted probability for the target class, while negative weights imply reduced probability. For instance, "Age" with an importance score of 10.09 significantly influences the grade prediction, favoring passing. Similarly, "Sponsorship Type" with an importance score of 5.67 positively affects the outcome.

The prediction probability section categorizes "PASS" and "FAIL." Positive and negative weights for these categories indicate feature impact on predictions. Attributes like {Age, Sponsorship Type}, and {Course, Country} impact predictions, while "FreqW1W7," "FreqW1," and "Gender" have opposing effects.

This study's findings underscore eLSTM's superiority over other tested time series approaches, attributed to its MSECosine loss function, offering improved training balance and gradient stability. These advantages elevate performance and generalization, differentiating eLSTM from other models.

Therefore, it can provide educators with a practical tool for early student grade predictions. The model, demonstrated in Figure. 30, helps identify students at risk, enabling targeted interventions. By using the proposed eLSTM that combines LSTM with MSECosine, insights into temporal dependencies and vanishing gradient issues can be gained, capturing nuanced performance patterns. Additionally, the feature selection process identifies influential factors, providing a holistic view of student success. Thus, institutions can integrate these findings into decision support systems for refined early warning and proactive interventions. These approaches consider factors like attendance, engagement, and historical academic performance, empowering informed intervention decisions.

VII. CONCLUSION AND FUTURE WORK

This study has extensively investigated the advantages of deep time series models over traditional methods for educational sequential data prediction. The utilization of RNNs, with their incorporation of nonlinear activation functions and feedback loops, has proven to be instrumental in capturing intricate nonlinear correlations among variables, distinguishing them from conventional time series models. However, a notable challenge encountered during the training phase of RNNs is the issue of vanishing or exploding gradients, which can hinder the model's convergence.

A significant contribution of this study lies in proposing a novel solution, namely the MSECosine loss function constructed through the combination of MSE and LogCosh. The aim of proposed MSECosine loss function is to address this limitation by providing more control over error magnification, particularly during suboptimal predictions. The amalgamation of these functions mitigates sudden spikes in error values, contributing a more stable training process. We utilise data on in higher education computer science program assessment and learning activities for early course grade prediction using the proposed solution, and compare this with benchmark approaches. This investigation is conducted by exploring the application of the models on 29 feature sets that are constructed based on several combination of demography, learning activities and assessment features.

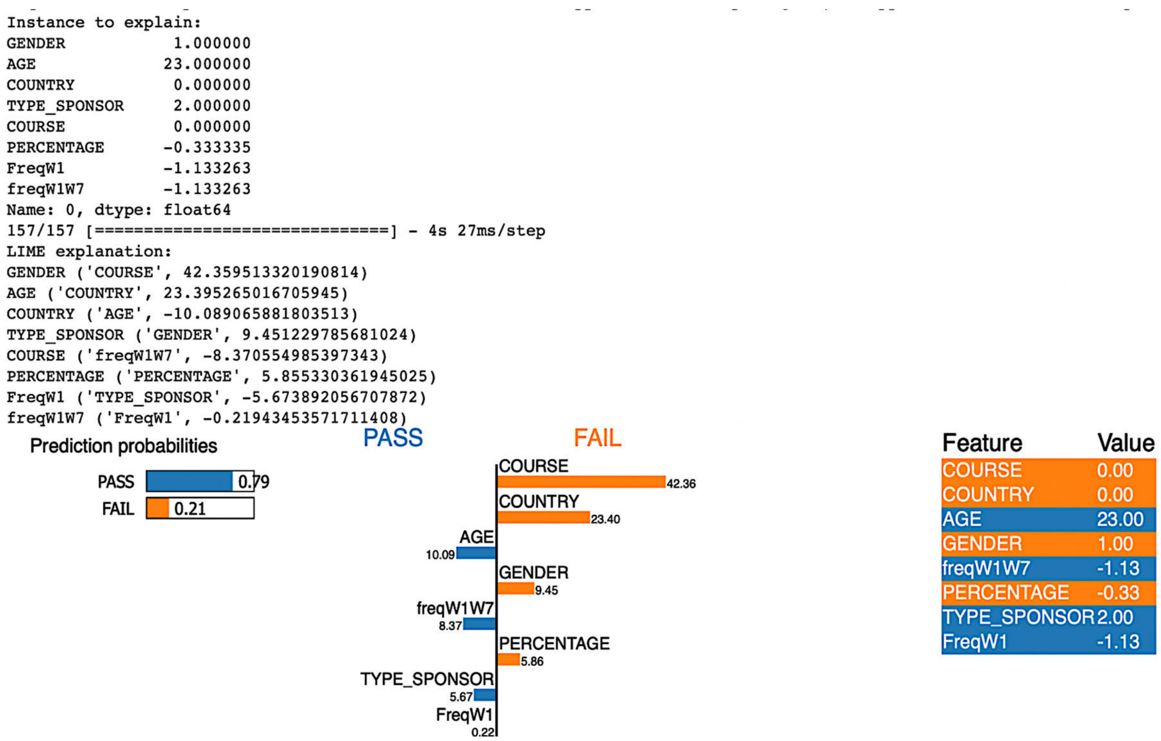
The culmination of Experiments 1, 2, and 3 holds significant relevance for predicting course grades in higher education computer science programs. Experiment 1 establishes a foundation by evaluating deep time series approaches on varied feature sets, while Experiment 2 refines the analysis by comparing the eLSTM method against basic approaches. Experiment 3, focusing on specific factors like demography, weekly engagement, and continuous assessment, adds granularity to the predictive models. These experiments collectively offer insights into the nuanced dynamics influencing academic performance, providing a holistic framework for the development of accurate and effective early course grade prediction models. This comprehensive approach is particularly vital in computer science programs, where diverse factors contribute to student success, aiding educators in tailored interventions and support strategies to enhance the overall learning experience.

Our research underscores the pivotal role of the proposed MSECosine loss function in effectively regulating errors during the training phase of LSTM models and enhancing the overall performance of deep time series approaches. By strategically combining the logarithmic term from LogCosh with the exponential terms of MSE, our approach offers a refined mechanism for error control. The eLSTM model, stemming from this methodology, surpasses other architectures in terms of accuracy and demonstrates lower error values. Specifically, our approach achieves an impressive 0.6191% accuracy with Feature Set 11, providing robust evidence of the superior performance of eLSTM for early grade prediction. In essence, our method not only introduces a novel approach for mitigating vanishing gradient issues but also attains the best results for early grade prediction, showcasing its effectiveness in improving both training dynamics and predictive accuracy. The performance of the proposed MSECosine loss function in mitigating the vanishing gradient issue is twice as effective as that of standard loss functions.

Moving forward, this research unlocks possibilities for further study in the realm of deep learning for time series data, and motivates works in early grade prediction. Future research endeavors could involve the refinement of existing loss functions, exploration of additional regularization techniques, and the incorporation of interpretability tools to enhance the transparency of model predictions. Additionally, the generalizability of the MSECosine loss function across various educational datasets and contexts warrants further investigation.

APPENDIX

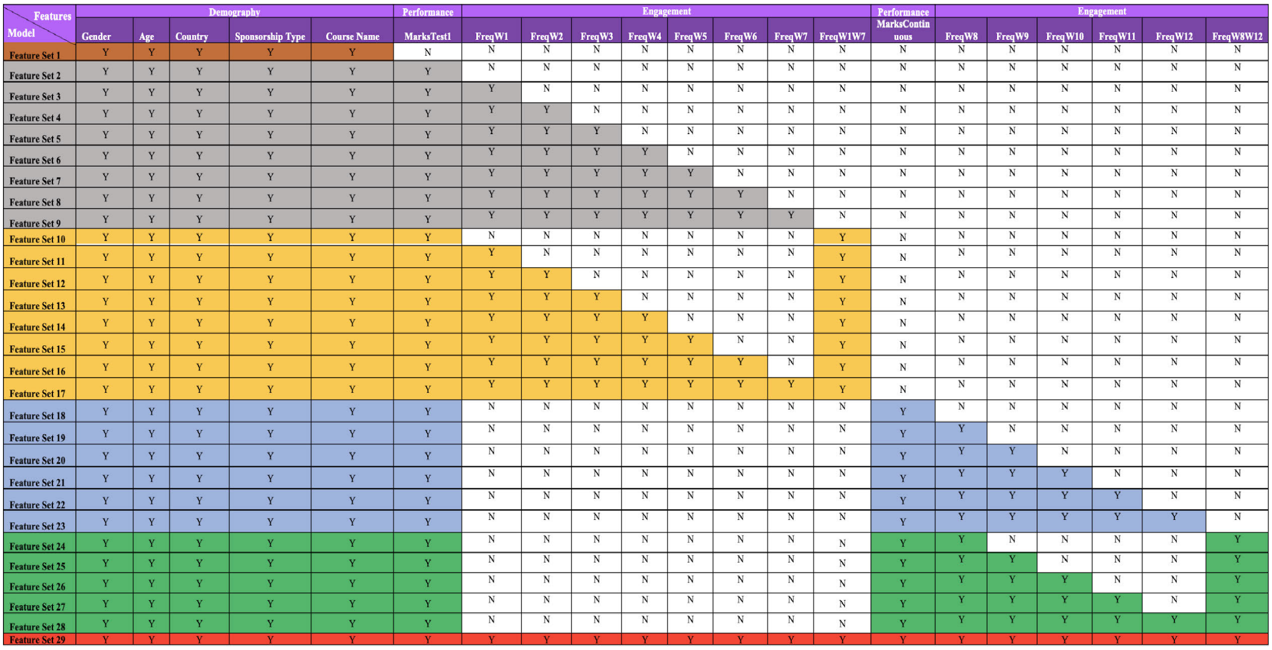
See Figures 30 and Figure 31 and Table 14–18.



```
Instance to explain:
GENDER            1.000000
AGE              23.000000
COUNTRY           0.000000
TYPE_SPONSOR      2.000000
COURSE            0.000000
PERCENTAGE       -0.333335
FreqW1           -1.133263
freqW1W7         -1.133263
Name: 0, dtype: float64
157/157 [==============================] - 4s 27ms/step
LIME explanation:
GENDER ('COURSE', 42.359513320190814)
AGE ('COUNTRY', 23.395265016705945)
COUNTRY ('AGE', -10.089065881803513)
TYPE_SPONSOR ('GENDER', 9.451229785681024)
COURSE ('freqW1W7', -8.370554985397343)
PERCENTAGE ('PERCENTAGE', 5.855330361945025)
FreqW1 ('TYPE_SPONSOR', -5.673892056707872)
freqW1W7 ('FreqW1', -0.21943453571711408)
```

Feature	Value
COURSE	0.00
COUNTRY	0.00
AGE	23.00
GENDER	1.00
freqW1W7	-1.13
PERCENTAGE	-0.33
TYPE_SPONSOR	2.00
FreqW1	-1.13

FIGURE 30. Visualizing the inner working of the proposed eLSTM models using LIME for precise early grade prediction.



Features / Model	Demography					Performance	Engagement								Performance	Engagement					
	Gender	Age	Country	Sponsorship Type	Course Name	MarksTest1	FreqW1	FreqW2	FreqW3	FreqW4	FreqW5	FreqW6	FreqW7	FreqW1W7	MarksContinuous	FreqW8	FreqW9	FreqW10	FreqW11	FreqW12	FreqW8W12
Feature Set 1	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Feature Set 2	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Feature Set 3	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Feature Set 4	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Feature Set 5	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N
Feature Set 6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N
Feature Set 7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
Feature Set 8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
Feature Set 9	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
Feature Set 10	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
Feature Set 11	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
Feature Set 12	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N
Feature Set 13	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N	N	N	N	N	N
Feature Set 14	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N	N
Feature Set 15	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	N	N	N	N	N
Feature Set 16	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	N	N	N	N	N	N
Feature Set 17	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N
Feature Set 18	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
Feature Set 19	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N
Feature Set 20	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	N
Feature Set 21	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	N
Feature Set 22	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	N
Feature Set 23	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N
Feature Set 24	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	Y
Feature Set 25	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y
Feature Set 26	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	Y
Feature Set 27	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
Feature Set 28	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
Feature Set 29	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

FIGURE 31. Construction of feature sets for student grade prediction models.

TABLE 14. **MLP performance on the twenty-nine designed feature set using the MSE loss function.**

MLP					
Feature Set	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss	RMSE
1	0.6330	0.5755	0.2275	0.2445	0.4944
2	0.6168	0.6004	0.2310	0.2396	0.4895
3	0.6365	0.5958	0.2242	0.2436	0.4935
4	0.6156	0.5856	0.2330	0.2403	0.4901
5	0.6425	0.5663	0.2244	0.2441	0.494
6	0.5414	0.6087	0.3849	0.2362	0.4753
7	0.5584	0.5700	0.2464	0.2429	0.4893
8	0.5122	0.5691	0.7304	0.2464	0.5016
9	0.5975	0.5727	0.2351	0.2427	0.4926
10	0.5975	0.5727	0.2351	0.2427	0.4926
11	0.5478	0.5571	0.5263	0.2503	0.4868
12	0.5880	0.6087	0.2384	0.2560	0.5508
13	0.6022	0.6133	0.2342	0.2376	0.4874
14	0.6002	0.6087	0.2351	0.2409	0.5224
15	0.6030	0.6087	0.2339	0.2517	0.5042
16	0.6081	0.6068	0.2339	0.2431	0.4930
17	0.6302	0.5921	0.2260	0.2421	0.4920
18	0.6042	0.6077	0.2337	0.2465	0.4912
19	0.5927	0.6087	0.2346	0.2772	0.5265
20	0.5758s	0.5930	0.2435	0.2422	0.4942
21	0.5967	0.6087	0.2390	0.2378	0.4939
22	0.5604	0.6031	0.2441	0.2384	0.5972
23	0.5762	0.6096	0.2437	0.2370	0.4919
24	0.6018	0.6077	0.2352	0.2383	0.5270
25	0.5923	0.6077	0.2361	0.2481	0.4980
26	0.5742	0.6087	0.2413	0.2499	0.4980
27	0.5770	0.5985	0.2421	0.2405	0.4933
28	0.5773	0.6077	0.2386	0.2385	0.5514
29	0.5635	0.6059	0.2432	0.2389	0.4931

TABLE 15. **CNN performance on the twenty-nine designed feature set using the MSE loss function.**

CNN					
Feature Set	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss	RMSE
1	0.5454	0.5994	0.4500	0.2431	0.5098
2	0.5971	0.5055	0.2340	0.2556	0.5055
3	0.5647	0.5967	0.2428	0.2409	0.4943
4	0.5912	0.6096	0.2390	0.2411	0.5043
5	0.5758	0.6059	0.2424	0.2367	0.4998
6	0.5722	0.6133	0.2412	0.2358	0.5782
7	0.5762	0.6013	0.2425	0.2394	0.5283
8	0.5730	0.5967	0.2419	0.2429	0.4999
9	0.5675	0.5893	0.2456	0.2425	0.5126
10	0.5655	0.6087	0.2449	0.2607	0.5228
11	0.5813	0.6068	0.2380	0.2486	0.4970
12	0.5659	0.6004	0.2442	0.2392	0.4934
13	0.5892	0.6087	0.2403	0.2460	0.5555
14	0.5722	0.6087	0.2438	0.2460	0.5072
15	0.5651	0.5866	0.2417	0.2391	0.4920
16	0.5813	0.6059	0.2412	0.2420	0.5029
17	0.5762	0.6087	0.2444	0.2536	0.5019
18	0.5635	0.6041	0.2414	0.2435	0.4992
19	0.5651	0.6096	0.2433	0.2424	0.5045
20	0.5912	0.6087	0.2402	0.2681	0.5403
21	0.6081	0.6123	0.2361	0.2425	0.5284
22	0.5371	0.5497	0.2549	0.2468	0.4984
23	0.5939	0.6031	0.2358	0.2395	0.4963
24	0.5647	0.6114	0.2436	0.2345	0.4957
25	0.5872	0.6022	0.2404	0.3097	0.4934
26	0.5655	0.6041	0.2448	0.2361	0.4934
27	0.5864	0.5939	0.2359	0.2407	0.4978
28	0.5813	0.6031	0.2369	0.2432	0.4980
29	0.6275	0.6041	0.2253	0.2432	0.5032

TABLE 16. LSTM performance on the twenty-nine designed feature set using the MSE loss function.

			LSTM		
Feature Set	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss	RMSE
1	0.5616	0.6133	0.2430	0.2393	0.5098
2	0.5655	0.6179	0.2429	0.2225	0.5055
3	0.5734	0.6022	0.2407	0.2367	0.4943
4	0.5604	0.6151	0.2410	0.2340	0.5043
5	0.5667	0.6123	0.2412	0.2355	0.4998
6	0.5620	0.6133	0.2434	0.2357	0.5782
7	0.5710	0.6206	0.2411	0.2346	0.5283
8	0.5608	0.6188	0.2420	0.2339	0.4999
9	0.5762	0.6169	0.2416	0.2346	0.5228
10	0.5762	0.6169	0.2416	0.2346	0.5228
11	0.5651	0.6225	0.2411	0.2356	0.4970
12	0.5714	0.6114	0.2421	0.2335	0.4934
13	0.5710	0.6179	0.2419	0.2346	0.5555
14	0.5718	0.6151	0.2430	0.2381	0.4920
15	0.5667	0.6123	0.2419	0.2341	0.5029
16	0.5667	0.6123	0.2419	0.2341	0.5029
17	0.5639	0.6041	0.2438	0.2361	0.5019
18	0.5635	0.6068	0.2451	0.2386	0.4991
19	0.5671	0.6031	0.2416	0.2356	0.5045
20	0.5734	0.6105	0.2423	0.2387	0.5403
21	0.5631	0.6133	0.2445	0.2373	0.5284
22	0.5335	0.6123	0.2549	0.2529	0.4984
23	0.5592	0.6151	0.2433	0.2336	0.4963
24	0.5612	0.6096	0.2431	0.2342	0.4957
25	0.5560	0.6151	0.2458	0.2377	0.5065
26	0.5691	0.6151	0.2423	0.2362	0.4934
27	0.5691	0.6151	0.2423	0.2362	0.4934
28	0.5734	0.6169	0.2420	0.2351	0.4978
29	0.5710	0.6123	0.2416	0.2353	0.4980

TABLE 17. CNN-LSTM performance on the twenty-nine designed feature set using the MSE loss function.

			CNN-LSTM		
Feature Set	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss	RMSE
1	0.5560	0.6041	0.2428	0.2356	0.4853
2	0.5683	0.6087	0.2433	0.2354	0.4852
3	0.5647	0.6087	0.2431	0.2349	0.4846
4	0.5651	0.6068	0.2420	0.2347	0.4844
5	0.5643	0.6077	0.2430	0.2357	0.4854
6	0.5620	0.6151	0.2433	0.2371	0.4854
7	0.5635	0.6105	0.2431	0.2391	0.4849
8	0.5651	0.6059	0.2427	0.2365	0.4863
9	0.5651	0.6059	0.2427	0.2365	0.4839
10	0.5596	0.6133	0.2445	0.2372	0.4869
11	0.5663	0.6169	0.2431	0.2369	0.4867
12	0.5659	0.6096	0.2433	0.2363	0.4860
13	0.5576	0.6114	0.2425	0.2345	0.4842

TABLE 17. ***(Continued.)*** **CNN-LSTM performance on the twenty-nine designed feature set using the MSE loss function.**

14	0.5631	0.6151	0.2425	0.2380	0.4860
15	0.5604	0.6068	0.2426	0.2390	0.4888
16	0.5596	0.6160	0.2423	0.2355	0.4852
17	0.5663	0.6169	0.2426	0.2364	0.4849
18	0.5616	0.6068	0.2436	0.2394	0.4893
19	0.5584	0.6123	0.2442	0.2373	0.4858
20	0.5588	0.6050	0.2432	0.2359	0.4857
21	0.5604	0.6142	0.2437	0.2384	0.4882
22	0.5584	0.6087	0.2434	0.2349	0.4846
23	0.5631	0.6142	0.2967	0.2431	0.4915
24	0.5679	0.6013	0.2433	0.2373	0.4929
25	0.5639	0.6059	0.2435	0.2421	0.4919
26	0.5655	0.6160	0.2437	0.2413	0.4919
27	0.5564	0.6179	0.2444	0.2417	0.4859
28	0.5588	0.6169	0.2435	0.2361	0.4859
29	0.5549	0.6114	0.2429	0.2354	0.4851

TABLE 18. **Proposed eLSTM performance on the twenty-nine designed feature selection models using the proposed MSECosine loss function.**

		Proposed eLSTM			
Feature Set	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss	RMSE
1	0.5651	0.6169	0.1793	0.1739	0.4848
2	0.5675	0.6179	0.1795	0.1735	0.4844
3	0.5726	0.6123	0.1791	0.1731	0.4841
4	0.5821	0.6225	0.1781	0.1733	0.4837
5	0.5821	0.6225	0.1781	0.1733	0.4850
6	0.5663	0.6169	0.1777	0.1725	0.4866
7	0.2140	0.6188	0.5142	0.1757	0.4893
8	0.5355	0.6215	0.1968	0.1727	0.4860
9	0.5635	0.6215	0.1803	0.1738	0.4871
10	0.5726	0.6188	0.1779	0.1724	0.4847
11	0.5643	0.6271	0.1787	0.1753	0.4835
12	0.5702	0.6206	0.1791	0.1740	0.4884
13	0.5714	0.6225	0.1790	0.1754	0.4844
14	0.5691	0.6225	0.1795	0.1730	0.4871
15	0.5671	0.6206	0.1790	0.1741	0.4838
16	0.5679	0.6160	0.1790	0.1730	0.4899
17	0.5647	0.6179	0.1797	0.1764	0.4862
18	0.5095	0.6151	0.2075	0.1752	0.4868
19	0.5616	0.6179	0.1799	0.1756	0.4839
20	0.5675	0.6169	0.1792	0.1754	0.4864
21	0.5706	0.6169	0.1782	0.1730	0.4863
22	0.5702	0.6151	0.1784	0.1745	0.4845
23	0.5695	0.6215	0.1783	0.1729	0.4935
24	0.5734	0.6179	0.1790	0.1728	0.4882
25	0.5742	0.6234	0.1786	0.1741	0.4855
26	0.5691	0.6160	0.1787	0.1729	0.4841
27	0.5612	0.6160	0.1777	0.1737	0.4871
28	0.5659	0.6206	0.1779	0.1730	0.4883
29	0.5722	0.6206	0.1788	0.1753	0.4844

ACKNOWLEDGMENT

The authors would like to express their gratitude to the project funder, the Airforce Office of Scientific Research (AOARD), USA, and its management office, and the Asian Office of Aerospace Research and Development (AOARD), Japan, for the opportunity. They would also like to thank the InfoComm Development Centre (IDEC), Universiti Putra Malaysia, for providing the data in this research.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] F. A. Al-Azazi and M. Ghurab, "ANN-LSTM: A deep learning model for early student performance prediction in MOOC," *Heliyon*, vol. 9, no. 4, Apr. 2023, Art. no. e15382, doi: 10.1016/j.heliyon.2023.e15382.

[2] R. A. N. Al-Tameemi, C. Johnson, R. Gitay, A.-S.-G. Abdel-Salam, K. A. Hazaa, A. BenSaid, and M. H. Romanowski, "Determinants of poor academic performance among undergraduate students—A systematic literature review," *Int. J. Educ. Res. Open*, vol. 4, Jan. 2023, Art. no. 100232, doi: 10.1016/j.ijedro.2023.100232.

[3] E. Alyahyan and D. Düştegör, "Predicting academic success in higher education: Literature review and best practices," *Int. J. Educ. Technol. Higher Educ.*, vol. 17, no. 1, p. 3, Dec. 2020, doi: 10.1186/s41239-020-0177-7.

[4] A. Al-Zawqari, D. Peumans, and G. Vandersteen, "A flexible feature selection approach for predicting students' academic performance in online courses," *Comput. Educ., Artif. Intell.*, vol. 3, Jan. 2022, Art. no. 100103, doi: 10.1016/j.caeai.2022.100103.

[5] Y. Baashar, G. Alkawsi, N. Ali, H. Alhussian, and H. T. Bahbouh, "Predicting student's performance using machine learning methods: A systematic literature review," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCOINS)*, Jul. 2021, pp. 357–362, doi: 10.1109/ICCOINS49721.2021.9497185.

[6] S. Rajendran, S. Chamundeswari, and A. A. Sinha, "Predicting the academic performance of middle- and high-school students using machine learning algorithms," *Social Sci. Humanities Open*, vol. 6, no. 1, 2022, Art. no. 100357, doi: 10.1016/j.ssaho.2022.100357.

[7] H. Sakız, F. Özdaş, AI. Göksu, and A. Ekinci, "A longitudinal analysis of academic achievement and its correlates in higher education," *SAGE Open*, vol. 11, no. 1, Jan. 2021, Art. no. 215824402110030, doi: 10.1177/21582440211003085.

[8] C. Smithikrai, T. Homklin, P. Pusapanich, V. Wongpinpech, and P. Kreausukon, "Factors influencing students' academic success: The mediating role of study engagement," *J. Behav. Sci.*, vol. 13, no. 1, pp. 1–14, 2018.

[9] P. Chaudhary and R. K. Singh, "A meta analysis of factors affecting teaching and student learning in higher education," *Frontiers Educ.*, vol. 6, Feb. 2022, Art. no. 824504, doi: 10.3389/feduc.2021.824504.

[10] M. Riestra-González, M. D. P. Paule-Ruíz, and F. Ortin, "Massive LMS log data analysis for the early prediction of course-agnostic student performance," *Comput. Educ.*, vol. 163, Apr. 2021, Art. no. 104108, doi: 10.1016/j.compedu.2020.104108.

[11] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944, doi: 10.1371/journal.pone.0180944.

[12] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Phil. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 379, no. 2194, Apr. 2021, Art. no. 20200209, doi: 10.1098/rsta.2020.0209.

[13] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An overview and comparison of supervised data mining techniques for student exam performance prediction," *Comput. Educ.*, vol. 143, Jan. 2020, Art. no. 103676, doi: 10.1016/j.compedu.2019.103676.

[14] K. Benidis, S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, F.-X. Aubet, L. Callot, and T. Januschowski, "Deep learning for time series forecasting: Tutorial and literature survey," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–36, Jul. 2023, doi: 10.1145/3533382.

[15] Z. Shen, Y. Zhang, J. Lu, J. Xu, and G. Xiao, "A novel time series forecasting model with deep learning," *Neurocomputing*, vol. 396, pp. 302–313, Jul. 2020, doi: 10.1016/j.neucom.2018.12.084.

[16] X. Wan, H. Liu, H. Xu, and X. Zhang, "Network traffic prediction based on LSTM and transfer learning," *IEEE Access*, vol. 10, pp. 86181–86190, 2022, doi: 10.1109/ACCESS.2022.3199372.

[17] U. M. Sirisha, M. C. Belavagi, and G. Attigeri, "Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison," *IEEE Access*, vol. 10, pp. 124715–124727, 2022, doi: 10.1109/ACCESS.2022.3224938.

[18] X. Wen and W. Li, "Time series prediction based on LSTM-attention-LSTM model," *IEEE Access*, vol. 11, pp. 48322–48331, 2023, doi: 10.1109/ACCESS.2023.3276628.

[19] X. Zhang, X. Liang, A. Zhiyuli, S. Zhang, R. Xu, and B. Wu, "AT-LSTM: An attention-based LSTM model for financial time series prediction," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 569, no. 5, Jul. 2019, Art. no. 052037, doi: 10.1088/1757-899X/569/5/052037.

[20] Y. K. Ee, N. M. Sharef, R. Yaakob, and K. A. Kasmiran, "LSTM based recurrent enhancement of DQN for stock trading," in *Proc. IEEE Conf. Big Data Anal. (ICBDA)*, Nov. 2020, pp. 38–44, doi: 10.1109/ICBDA50157.2020.9289832.

[21] C.-N. Wang, F.-C. Yang, T. M. N. Vo, V. T. T. Nguyen, and M. Singh, "Enhancing efficiency and cost-effectiveness: A groundbreaking bi-algorithm MCDM approach," *Appl. Sci.*, vol. 13, no. 16, p. 9105, Aug. 2023, doi: 10.3390/app13169105.

[22] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.

[23] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity*, vol. 2021, pp. 1–10, Jul. 2021, doi: 10.1155/2021/9958203.

[24] A. Ghazvini, S. N. H. S. Abdullah, M. K. Hasan, and D. Z. A. B. Kasim, "Crime spatiotemporal prediction with fused objective function in time delay neural network," *IEEE Access*, vol. 8, pp. 115167–115183, 2020, doi: 10.1109/ACCESS.2020.3002766.

[25] Y. Guo, Z. Wu, and Y. Ji, "A hybrid deep representation learning model for time series classification and prediction," in *Proc. 3rd Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2017, pp. 226–231, doi: 10.1109/BIGCOM.2017.13.

[26] Y. Chen, K. He, and G. K. F. Tso, "Forecasting crude oil prices: A deep learning based model," *Proc. Comput. Sci.*, vol. 122, pp. 300–307, Jan. 2017, doi: 10.1016/j.procs.2017.11.373.

[27] K. He, Q. Yang, L. Ji, J. Pan, and Y. Zou, "Financial time series forecasting with the deep learning ensemble model," *Mathematics*, vol. 11, no. 4, p. 1054, Feb. 2023, doi: 10.3390/math11041054.

[28] J. Jiang, L. Wu, H. Zhao, H. Zhu, and W. Zhang, "Forecasting movements of stock time series based on hidden state guided deep learning approach," *Inf. Process. Manage.*, vol. 60, no. 3, May 2023, Art. no. 103328, doi: 10.1016/j.ipm.2023.103328.

[29] S. Zhou, C. Wei, C. Song, Y. Fu, R. Luo, W. Chang, and L. Yang, "A hybrid deep learning model for short-term traffic flow pre-diction considering spatiotemporal features," *Sustainability*, vol. 14, no. 16, p. 10039, Aug. 2022, doi: 10.3390/su141610039.

[30] M. Windarti and P. T. Prasetyaninrum, "Prediction analysis student graduate using multilayer perceptron," in *Proc. Int. Conf. Online Blended Learn. (ICOBL)*, 2020, doi: 10.2991/assehr.k.200521.011.

[31] S. Yang, "A novel study on deep learning framework to predict and analyze the financial time series information," *Future Gener. Comput. Syst.*, vol. 125, pp. 812–819, Dec. 2021, doi: 10.1016/j.future.2021.07.017.

[32] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022, doi: 10.1109/TITS.2021.3054840.

[33] Z. Liang, J. Mao, K. Lu, Z. Ba, and G. Li, "Combining deep neural network and bibliometric indicator for emerging research topic prediction," *Inf. Process. Manage.*, vol. 58, no. 5, Sep. 2021, Art. no. 102611, doi: 10.1016/j.ipm.2021.102611.

[34] A. Dutta, G. Pooja, N. Jain, R. R. Panda, and N. K. Nagwani, "A hybrid deep learning approach for stock price prediction," in *Machine Learning for Predictive Analysis*, vol. 141, A. Joshi, M. Khosravy, N. Gupta, Eds. Singapore: Springer, 2021, pp. 1–10, doi: 10.1007/978-981-15-7106-0_1.

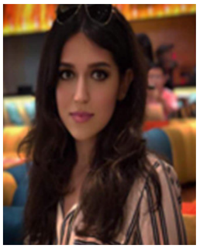
[35] A. M. Rather, "LSTM-based deep learning model for stock prediction and predictive optimization model," *EURO J. Decis. Processes*, vol. 9, Jan. 2021, Art. no. 100001, doi: 10.1016/j.ejdp.2021.100001.

[36] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Mach. Learn. With Appl.*, vol. 9, Sep. 2022, Art. no. 100320, doi: 10.1016/j.mlwa.2022.100320.

[37] S. Thapa, Z. Zhao, B. Li, L. Lu, D. Fu, X. Shi, B. Tang, and H. Qi, "Snowmelt-driven streamflow prediction using machine learning techniques (LSTM, NARX, GPR, and SVR)," *Water*, vol. 12, no. 6, p. 1734, Jun. 2020, doi: 10.3390/w12061734.

[38] B. Ngwira, B. Gobin-Rahimbux, and N. G. Sahib, "A deep-learning framework for analysing students' review in higher education," *Comput. Intell. Neurosci.*, vol. 2023, pp. 1–13, Mar. 2023, doi: 10.1155/2023/8462575.

[39] A. A. Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani, "Prediction of student course selection in online higher education institutes using neural network," *Comput. Educ*, vol. 65, pp. 1–11, Jul. 2013, doi: 10.1016/j.compedu.2013.01.015.



ANAHITA GHAZVINI (Member, IEEE) received the B.Sc. degree (Hons.) in information technology (computer science), the M.Sc. degree in information technology (artificial intelligence), and the Ph.D. degree in computer science from the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM), in 2013, 2016, and 2022, respectively. She is currently a Postdoctoral Researcher with the Faculty of Computer Science and Technology (FSKTM), Universiti Putra Malaysia (UPM). She has published many peer-reviewed journal articles. Her research interests include artificial intelligence, algorithms, distributed computing, cyber forensics, learning analytics, and e-learning.

NURFADHLINA MOHD SHAREF (Senior Member, IEEE) is currently an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Malaysia. Her works are applied to education, biodiversity, agriculture, and health. Her research interests include artificial intelligence, machine learning, and data science. She has special interests in learning analytics and e-learning. She is one of the national task force members in AI and focuses on talent development and ethics.

FATIMAH BINTI SIDI (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science and the Ph.D. degree in management information systems from Universiti Putra Malaysia (UPM), Malaysia, with a focus on the transformation of extracted knowledge in Malay unstructured documents into an interrogative structured form. She is currently the Director of the Infocomm Development Centre (iDEC) and an Associate Professor of computer science with the Department of Computer Science, Faculty of Computer Science and Information Technology (CSiT), UPM. She serves several research projects funded by the Institute and the Ministry of Higher Education. Her current research interests include big data analytics, data quality, knowledge and information management systems, data and knowledge engineering, databases, and data warehouses.