INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 4, PP. 1-10
Manuscript received August 30, 2025; revised September 2025. doi: 10.24425/ijet.2025.155475

Enhancing hyperparameters of LSTM network models through Genetic Algorithm for Virtual Learning Environment prediction

Edi Ismanto, Hadhrami Ab Ghani, and Nurul Izrin Md Saleh

***Abstract*—In today's technology-driven era, innovative methods for predicting behaviors and patterns are crucial. Virtual Learning Environments (VLEs) represent a rich domain for exploration due to their abundant data and potential for enhancing learning experiences. Long Short-Term Memory (LSTM) models, while proficient with sequential data, face challenges such as overfitting and gradient issues. This study investigates the optimization of LSTM parameters and hyperparameters for VLE prediction. Adaptive gradient-based algorithms, including ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX, exhibited superior performance. The LSTM model with ADADELTA achieved 91% accuracy for BBB course data, while ADAGRAD LSTM models attained average accuracies of 80% and 85% for DDD and FFF courses, respectively. Genetic algorithms for hyperparameter optimization significantly contributed, with the GA + LSTM + ADAGRAD model achieving 88% and 87% accuracy in the 7th and 9th models for BBB course data. The GA + LSTM + ADADELTA model produced average accuracy rates of 80% and 84% in DDD and FFF course data, with the highest accuracy rates of 86% and 93%, as well. These findings highlight the effectiveness of adaptive and genetic algorithms in enhancing LSTM model performance for VLE prediction, offering valuable insights for educational technology advancement.**

***Keywords*—long short-term memory; genetic algorithm; adaptive gradient-based; hyperparameter optimization; virtual learning environment**

I. Introduction

In this ever-evolving era of information technology, which deftly steers various facets of life, changes including in behaviors and patterns unfold seamlessly as time passes. Navigating these temporal changes require innovative and efficient approaches to predicting myriad essential patterns and classes is highly desirable. One intriguing predictive-based research field for exploration is virtual learning environments, where critical academic performance patterns can be generated and analyzed to enhance the learning experience [1]. This study utilizes LSTM (Long Short-Term Memory) to optimize predictions within virtual learning environments. For modeling sequential data, such as the order of events in virtual learning environments (VLE), LSTM is an effective tool [2]. While LSTM generally handles sequential data well, the model is susceptible to overfitting, especially when exposed to complex data [3]. Moreover, the problems of decreasing gradient and bursting gradient often affect LSTM. The issue of vanishing gradients arises when the neural network's gradient multiplied by its weight gets extremely small, resulting in sluggish or unstable learning [4]. On the other hand, the exploding gradient problem arises when the gradient grows exponentially, which can also disrupt the learning process and make it unstable [5].

Therefore, optimizing LSTM becomes crucial to address these various issues. Through LSTM optimization, we can lower the likelihood of overfitting, increase learning speed and stability, and improve the capability of the design to recognize complex designs in sequential data. This will enable more effective and efficient use of LSTM in various applications, including predictions in virtual learning environments and other fields that leverage sequential data. This study employs two optimization strategies—parameter optimization with an adaptive gradient-based algorithm and hyperparameter optimization with a genetic algorithm—to enhance LSTM performance.

Optimizing LSTM parameters with adaptive gradient-based schemes offers efficient learning [6]. These algorithms dynamically adjust learning rates for each parameter, ensuring stable learning even with complex data [7]. These algorithms facilitate faster convergence and prevent gradient-related issues, such as vanishing or exploding gradients [8]. Their adaptability allows LSTM models to dynamically adjust to data variations [9]. Likewise, genetic algorithms provide an effective global solution search for LSTM hyperparameter optimization while maintaining parameter space diversity [10]. The algorithm is adaptive, scalable, and flexible, and it keeps getting better at solving problems [11]. With the use of this algorithm, it is possible to determine the ideal set of hyperparameters, producing an LSTM model for sequential data prediction that is more precise and effective.

The purpose of this research is to evaluate two theories about the optimization of hyperparameters and parameters in LSTM network models for virtual learning environment prediction. Firstly, the hypothesis suggests that the use of adaptive gradient-based algorithms will result in LSTM models that are more stable, converge faster, and can overcome gradient issues like vanishing or exploding gradients [12]. With the adaptability of these algorithms, it is anticipated that LSTM models optimized

Edi Ismanto is with Department of Informatics, Universitas Muhammadiyah Riau, Indonesia (e-mail: edi.ismanto@umri.ac.id).
Hadhrami Bin Ab Ghani is with Faculty of Data Science and Computing at Universiti Kelantan (UMK), Malaysia (e-mail: hadhrami.ag@umk.edu.my).
Nurul Izrin Binti MD Saleh is with Faculty of Computing , Universiti Teknikal Malaysia Melaka, Malaysia (email: nurulizrin.ms@utem.edu.my).

with this approach will perform better in predicting behaviors and patterns within virtual learning environments. Secondly, another hypothesis posits that utilizing genetic algorithms to optimize LSTM hyperparameters will yield more optimal and adaptive configurations. Considering the complexity and variability of data within virtual learning environments [13], genetic algorithms can discover better combinations of hyperparameters. Consequently, LSTM models optimized with genetic algorithms are expected to provide more accurate and responsive predictions to changes in the learning environment. Through this research, it is expected to gain a better understanding of effective approaches in optimizing LSTM models for predicting virtual learning environments, thus contributing to the advancement of digital learning technology.

A. Related Work

We reviewed the literature on subjects relevant to our research. Finding advancements in prediction models for deep learning-based online learning, particularly about greatly accessible distance education (MOOCs) or virtual classrooms (VLEs), is the aim of the literature review. As a result, the findings of the literature review can be used to contrast or compare our research with other studies.

In her research,[14] proposes the ANN-LSTM, a popular multi-class model that uses ANN (Artificial Networks) and LSTM (Short-term Memory) to predict the achievement of students. The findings show that the ANN-LSTM model is superior to the baseline models in terms of effectiveness. ANN-LSTM achieves an accuracy of roughly 70%. In [15], a suggested approach is a customized recommendation system based on the MOOC system. Some corresponding strategies are presented in [1] to improve the recommendation method's accuracy, which is by the encoder with two-channel illustrations from the Transformers (BERT) model. The results of the experiment demonstrate that the suggested model produces recommendation results with the same level of effectiveness as alternative approaches. In this work, [16] uses a range of automatic learning methods on open datasets, such as different kinds of Artificial Neural Networks (ANN) and tree-based models, to investigate the elements that impact the learning process in VLE platforms. Another course recommendation system using deep neural network algorithms has also been designed using the neural collaborative filtering (NCF) approach [17]. It is reported that the NCF model outperforms the cooperative filtering (CF) model by 57.7% in terms of the RMSE achievement recorded.

In [18], an adversarial network called the Sequential Conditional Generative Adversarial Network (SC-GAN) was used. It summarises each student's prior behavior. The corresponding results have indicated that the proposed SCGAN outperformed the standard up-sampling methods. Specifically, compared to Random Over-sampling, the SCGAN demonstrated an improved AUC of 7.07%. Based on the information about student behaviors, [19] has offered a strategy for predicting school dropout rates that use a pipeline model known as CLSA. Local features are extracted by the CLSA model using LSTM and CNN. 87.6% accuracy was attained by the model in tests conducted on the KDD 2015 data set. Other predictive models such as in [20] and [21] have employed

The KDD 2015 dataset, which are trained and tested with CNN and LSTM models incorporated with bagging techniques to attain average accuracy values of around 91%. In another paper, [16], an innovative method is suggested that makes use of a hyper-model called CONV-LSTM, which blends a CNN and LSTM to come in instinctively compile features from MOOC raw data and forecast whether or not each student will finish the course. Regarding performance outcomes, the suggested model is superior to the standard approach. With an accuracy of 84.57%, the LSTM has the highest predictive power when it comes to differentiating between students who pass and those who fail when compared to all other options [22].

In [5], students' cognitive states are classified using a novel GA-CNN (Graph-based conventional attention neural networks) architecture. Compared to alternative approaches, classification accuracy has been increased dramatically to around 87%. This study looks at the brain Chabot's signals and interactions to develop a new model for predicting how students will behave in online courses [23]. The techniques for extracting features are CNN and RB-RNN (Radian Basis-Recurrent Neural Networks). When compared to the basic method, the accuracy results demonstrate a significant improvement. To simulate and forecast student dropout behavior, [18] suggests using the PMCT stands for Parallel Multiscale Convolutional Temporal architecture. The outcomes of the experiment demonstrate that the suggested model has improved its prediction accuracy than the baseline method using two sizable datasets. Another LSTM-based scheme, [21], has been proposed to predict the time of subsequent interactions as well as the user's experience of those interactions. According to the study, the model's performance can be significantly increased by accounting for the correlation between an action taken by a user and when it happens. Additionally, the prediction results can be used to examine online learning behaviors and dropout rates.

The use of a DL (Deep learning) model, specifically the LSTM algorithm, is the commonality between the research to be done and earlier studies. The LSTM performs well in time series data prediction in earlier research. This study differs from others in that it uses genetic algorithms (GA) and adaptive gradient-based algorithms to optimize parameters and hyperparameters, something that was not done in earlier studies.

II. METHODOLOGY

We provide a framework for forecasting academic achievement to address the aforementioned issues by combining knowledge-based data with behavioral and learning data. The objective is to offer more accurate forecasts, particularly for online learning, in order to reduce student failure.

In order to do this, we use course data from students, which we can access from the time they enroll in classes until they graduate, enabling us to observe how they learn. The perspective domain is also used to cluster the knowledge-based data and add them to the forecasting procedure.

A. Data Pre-processing

During the data preparation stage, the prediction-related features were extracted from the OULA data [8], [4]. The data

for OULA courses is shown in Fig. 1. Presentation codes BBB, DDD, and FFF were used in this study since these three courses have complete data up until the end of the lecture. The dataset is preprocessed to select features that will be used to train and test the model. The features that have been chosen and will be put to use are the module code, presentation code, student ID, clicks, assignment assessment, average assignment assessment, and final results.

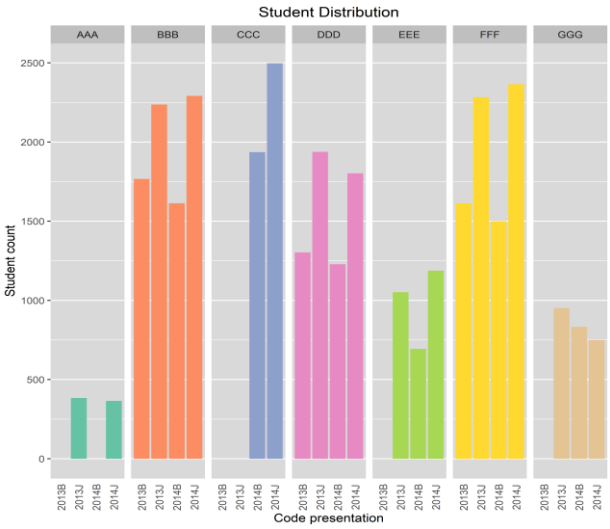

Fig. 1. Course information.

There are two presentation codes or semester codes in the BBB, DDD, and FFF courses: "B" begins in February, while "J" begins in October. The BBB, DDD, and FFF courses will be divided into sixty percent of the data for instruction, twenty percent for verification, and twenty percent for evaluation.

B. The LSTM's Architecture

An ANN (Artificial Neural Network) algorithm known as LSTM was created to address the "enduring memory" issue. The shortcomings of the RNN (Recurrent Neural Networks) algorithm are addressed by LSTM [13]. One area where RNNs struggle is remembering details in lengthy data sequences. Because of their distinctive memory unit design, LSTMs can handle this problem more successfully [3]. Fig. 2 displays the proposed LSTM architectural model from this study.

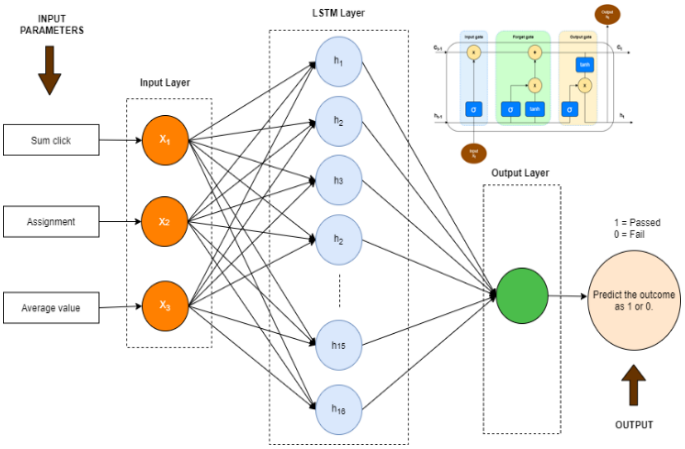

Fig. 2. The proposed LSTM architecture model.

One crucial stage in the development of an LSTM model that greatly affects the success and performance of the model is hyperparameter optimization [2]. Optimizing hyperparameters is a crucial stage in creating robust and efficient machine-learning models. This enables the model to achieve better performance, minimize overfitting risk, and optimize computer memory utilization, resulting in better results across various tasks and applications [24]. The LSTM architecture has several hyperparameters that affect its performance, including LSTM units in total (neurons), number of LSTM layers, learning rate, batch size, sequencing parameters, dropout rate, L2 or L1 regularisation, activation function, optimizer, weight initialization, gate usage (gates), loss function, epochs, and early stopping [25]. The following equations describe the functions of a LSTM cell:

Forget the Entrance:

$$f_t = \sigma\left(w_f \circ \begin{bmatrix} h_{t-1} \\ X_t \end{bmatrix} + b_f\right) \quad (1)$$

A gate for input:

$$I_t = \sigma\left(w_i \circ \begin{bmatrix} h_{t-1} \\ X_t \end{bmatrix} + b_i\right) \quad (2)$$

Cell State:

$$\tilde{C}_t = \tanh(w_c . [h_{t-1}, x_t] + b_c) \quad (3)$$

Update Cell State:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Output Gate:

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

Hidden State:

$$h_t = O_t * \tanh(C_t) \quad (6)$$

The LSTM formulas represent the mathematical operations that occur within an LSTM cell, which is utilized in neural networks to understand and model sequential data. The key components within these formulas include the disregard gate(f_t), input gate(i_t), condition of the cell(C_t), update of the cell, output gate(o_t), and concealed state(h_t). The memory gate regulates the amount of information which is disregarded based on the earlier cell state. The amount of new data added to the input gates determines the state of the cell. The actual cell state at a given time is the outcome of mixing the earlier state of cells with information filtered through the disregard gate and the estimated new cell state (C_t) filtered through the input gate. The gate for output ascertains the cell state's quantity that is utilized to create the concealed state, which is frequently output or used in particular sequential modeling tasks. These formulas illustrate how an LSTM cell processes information from one

time step to the next within a sequence, with a specific ability to handle issues with vanishing gradients and preserve long-term retention in sequential information.

C. Adaptive Gradient-based (AGb) Algorithms

The AGb algorithm is an optimization method used to find the optimal value in mathematical optimization problems [20]. This algorithm's primary feature is its capacity to automatically modify the learning rate while it is optimizing [26]. This adaptive gradient-based algorithm helps the optimization process to better converge when dealing with complex problems that have large parameter spaces [11]. The weights (parameters) in a model can be changed using an adaptive gradient-based algorithm, which permits the model to learn from data and generate more precise predictions [27]. Several well-known algorithms based on adaptive gradients are Adam, Nadam, Adagrad, Adadelta, and Adamax [26]. In this study, five Adaptive Gradient-based algorithms are used to optimize the parameters of the LSTM architecture. The five Adaptive Gradient-based algorithms' formulas are as follows:

Adam algorithm formulas [26]**:**

Momentum update:

$$m_t = \beta_1 * m_{(t-1)} + (1 - \beta_1) * g_t \quad (7)$$

Variance momentum update:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) * (g_t)^2 \quad (8)$$

First-moment bias correction:

$$\acute{m}_t = \frac{m_t}{1-\beta_1^t} \quad (9)$$

Second-moment bias correction:

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (10)$$

Parameter update:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} . \hat{m}_t \quad (11)$$

Nadam algorithm formulas [26]**:**

Momentum update:

$$m_t = \beta_1 * m_{(t-1)} + (1 - \beta_1) * g_t \quad (12)$$

Variance momentum update:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) * (g_t)^2 \quad (13)$$

First-moment bias correction:

$$\acute{m}_t = \frac{m_t}{1-\beta_1^t} \quad (14)$$

Second-moment bias correction:

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (15)$$

Nesterov Parameter Update:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} . (\beta_1 . \hat{m}_t + (1 - \beta_1) . g_t) \quad (16)$$

Adagrad algorithm formulas [19]**:**

Gradient accumulation update:

$$G_t = G_{t-1} + g_t^2 \quad (17)$$

Parameter update:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{G_t + \epsilon}} . g_t \quad (18)$$

Adadelta algorithm formulas [26]**:**

The mean of squared gradients exponentially:

$$e[g^2]_t = \rho . e[g^2]_{t-1} + (1 - \rho) . g_t^2 \quad (19)$$

Delta parameter (update on the parameter):

$$\Delta\theta_t = - \frac{\sqrt{e[\Delta\theta^2]_{t-1} + \epsilon}}{\sqrt{e[g^2]_t + \epsilon}} . g_t \quad (20)$$

Parameter update:

$$\theta_t = \theta_{t-1} + \Delta\theta_t \quad (21)$$

Adamax algorithm formulas [26]**:**

Variance momentum update:

$$v_t = \max(\beta_2 . v_{t-1}, |g_t|) \quad (22)$$

Parameter update:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{v_t} . g_t \quad (23)$$

Adam, Nadam, Adagrad, Adadelta, and Adamax are optimization algorithms used in ML (Machine learning). In the context of these algorithms, common symbols include m_t for the first moment of the gradient, β_1 and β_2 for adjustment factors, g_t for the gradient at a given iteration, v_t for the second moment of the gradient, α for the $learning\ rate$, ϵ for a modest amount to avoid division by zero, and additional terms specific to each algorithm. For example, Nadam incorporates Nesterov acceleration, while Adagrad accumulates gradients over time. Adadelta utilizes exponentially moving averages, and Adamax calculates the second moment based on the gradient's absolute

value. By modifying the learning rate and monitoring gradient information to enhance convergence, these algorithms aid in the optimization of model parameters during training.

D. Genetics Algorithms (GA)

The GA are computational techniques inspired by natural selection and evolution in genetics [25]. Natural selection, crossover, mutation, and reproduction are evolutionary processes that genetic algorithms (GAs) are modeled after [28]. GA can handle complex search space problems, find global solutions, be flexible and parallel, tolerate noisy data, have a wide range of applications, and be capable of exploration and exploitation [15]. The genetic algorithm pseudocode used to optimize the LSTM model is shown in Table I.

TABLE I
GA-LSTM PSEUDOCODE

	Algorithm 1: Pseudocode LSTM with Genetic Algorithm
0:	**START**
1:	**Input:**
2:	Dataset: $X_{train}, Y_{train}, X_{test}, Y_{test}$
3:	Initial Hyperparameters: $units, weights, bias, dropout, learning_{rate}, batch\_size, epochs, loss, metrics$
4:	Optimizers:[$'adam', 'nadam', 'adadelta', 'adagrad', 'adamax'$]
5:	i: Integer; fitness: Real; Population: Population; Best Individual: Individual;
6:	**Process:**
7:	SetLength(Pop, Size);
8:	for i := 0 to Size - 1 do
9:	begin
10:	Pop[i].units := Random(100) + 1;
11:	Pop[i].dropout := Random;
12:	Pop[i].learning_rate := Random;
13:	Pop[i].batch_size := Random(100) + 1;
14:	Pop[i].epochs := Random(100) + 1;
15:	Pop[i].optimizer:= $'adam', 'nadam', 'adadelta', 'adagrad', 'adamax'$]';
16:	
17:	**Output:**
18:	Function Evaluate LSTM(Indiv: Individual): Real;
19:	Best Individual := Select Best Individual (Population);
20:	**END**

III. EXPERIMENTS

A. Dataset

The experiment's student data came from OULA Data, which is sourced from the UK's Open University. The dataset consists of details on 32.593 students enrolled in 22 classes, evaluation outcomes, and recordings of their discussions with the Virtual Learning Environment (VLE), summarized daily by clicking counts (10.655.280 entries). The VLE information makes it possible to examine course design from the perspective of learning, and the data itself can be used to assess how much of an impact VLE has on learning objectives.

B. Model Evaluation

In this study, the Genetic algorithm and the adaptive gradient-based algorithm were utilized to optimize the LSTM model's parameters and hyperparameters to forecast student performance using OULA data. The LSTM model predicts whether a student will pass or fail each semester. Furthermore, we evaluate the effectiveness of every optimization algorithm on OULA data by dividing it into 10 deciles using three different course datasets.

Using an 80:20 ratio, test, and training data were extracted from each dataset for the forecasting process. The LSTM model developed for predicting student performance uses 3 (three) input layers, 2 (two) output layers with 1 node and a sigmoid activation function, 1 (one) hidden layer with 16 nodes, and a hyperbolic tangent activation function, which is used to solve the function non-linear. For the activation of the input layer, use a value of 0, and for the standard deviation use a value of 1. The LSTM model is then enhanced by a dropout layer, which is set to 50 % in each training step to prevent overfitting. The LSTM model was trained using batch size 32, a learning rate of 10%, and epoch 50 with the back-propagation method.

In this study, four distinct metrics were employed. The primary metric used to assess the prediction models' performance was accuracy. Given that the binary classification method is used by the model to predict, the accuracy can be described as follows [29]: $True\ Positive$ (TP): The quantity of positive samples that the model accurately classifies as positive. $True\ Negative$ (TN): The quantity of negative samples that the model accurately classifies as negative. $False\ Positive$ (FP): The quantity of negative samples that the model mistakenly classifies as positive. $False\ Negative$ (FN): The quantity of positive samples that the model misclassifies as negative. The following formula is used to determine the accuracy [30]:

$$Accuracy = \frac{N-(TP+TN)}{N\ (Total\ number\ of\ observations)} \quad (24)$$

In Equation 24, TP represents$True\ Positive$, TN represents$True\ Negative$, FP represents$False\ Positive$, and FN represents$False\ Negative$. Using additional metrics, like F-score, recall, and preciseness, the model's efficacy was evaluated, which is elaborated upon below.
The Precision is calculated using the formula [30]:

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (25)$$

The Recall is calculated using the formula:

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (26)$$

In Equation 25, $True\ Positive$ represents how many positive samples were accurately classified, and $False\ Positive$ shows the amount of adverse samples that were incorrectly identified as positive. In Equation 26, $True\ Positive$ represents the number of properly identified samples with positive classification, and $False\ Negative$ represents the quantity of positive samples that were mistakenly labeled as negative. The F1 Score is calculated using the equation below [30]:

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision\ x\ Recall} \quad (27)$$

In Equation 27, Precision and Recall are calculated using the formulas given in Equations 25 and 26.

IV. Results and Discussion

Algorithm models for LSTM were put to the test, and their performance was evaluated. Three factors are taken into account to determine whether a student will pass or fail the course: How many times the virtual learning environment has been clicked, the number of assessments submitted, and the average assessment grade. The length of each course varies, so the course data is divided into eleven deciles. Table II displays the information that was employed to assess and instruct the models.

TABLE II
Data For Training and Testing

Training data	Total data	Testing data	Total data
BBB data	3.858	BBB data	1.521
DDD data	2.830	DDD data	1.150
FFF data	3.818	FFF data	1.503

A. Performance Results for the LSTM Model Using the Adaptive Gradient-Based Optimization

To create a more accurate model, the LSTM model is trained and tested using the Adaptive Gradient-Based Optimization algorithm. The Adaptive Gradient-Based Optimization algorithm is used to optimize the model parameters to fit the training data by determining the minimum (or maximum) value of a cost or loss function. Model parameters are weights and biases that are adjusted in a machine learning model during training to reduce the cost function or loss function. The Adaptive Gradient-Based Optimization algorithm is a process that takes place during the backpropagation phase. The most popular Adaptive Gradient-Based Optimization algorithms are ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX. This algorithm will be used to modify the LSTM model's parameters in order to achieve the best predictive model performance.

Table III displays the outcomes of applying Adaptive Gradient-based algorithms to optimize the LSTM model parameters for the BBB Course data. Table III displays the accuracy and recall performance that were caused by the LSTM model.

TABLE III
The LSTM Model's Performance After parameter optimization with an Adaptive Gradient-based algorithm on the BBB Course

Combining the LSTM Model and AGB (Adaptive Gradient-based) Algorithms					
	Accuracy				
Models	ADAM	NADAM	ADADELTA	ADAGRAD	ADAMAX
0	0.75	0.75	0.75	0.75	0.75
1	0.60	0.59	0.75	0.58	0.59
2	0.69	0.65	0.78	0.69	0.60
3	0.58	0.69	0.79	0.57	0.73
4	0.78	0.73	0.80	0.73	0.76
5	0.79	0.83	0.84	0.78	0.74
6	0.87	0.87	0.85	0.79	0.87
7	0.87	0.89	0.87	0.85	0.83
8	0.87	0.71	0.78	0.64	0.69
9	0.91	0.87	**0.90**	0.92	0.91
10	0.91	0.91	**0.91**	0.92	0.91
Average	0.78	0.77	**0.82**	0.75	0.76

The accuracy and recall values of the LSTM model in the BBB Course significantly improved after parameter optimization using Five Adaptive Gradient-based algorithms: ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX. To see how well the Five Adaptive Gradient-based algorithms performed on the BBB course data, the average accuracy and recall of each decile/model are computed. The LSTM model with ADADELTA optimization produced the best model accuracy and recall values on the BBB course data, with an average model accuracy value of 82%. The results of optimizing the LSTM model parameters for the DDD Course data using the AGB (Adaptive Gradient-based) algorithms are shown in Table IV.

TABLE IV
The LSTM Model's Performance After parameter optimization with an Adaptive Gradient-based algorithm on the DDD Course

Combining the LSTM Model and AGB (Adaptive Gradient-based) Algorithms					
	Accuracy				
Models	ADAM	NADAM	ADADELTA	ADAGRAD	ADAMAX
0	0.69	0.69	0.68	0.69	0.69
1	0.74	0.74	0.71	0.73	0.75
2	0.76	0.77	0.72	0.77	0.78
3	0.77	0.78	0.76	0.80	0.79
4	0.78	0.77	0.76	0.79	0.77
5	0.81	0.80	0.75	0.81	0.80
6	0.80	0.80	0.77	0.83	0.83
7	0.81	0.80	0.80	0.83	0.81
8	0.85	0.85	0.81	0.84	0.84
9	0.85	0.85	0.84	**0.87**	0.86
10	0.86	0.86	0.85	0.86	0.86
Average	0.79	0.79	0.77	**0.80**	0.80

The accuracy and recall values of the LSTM model in the DDD Course significantly improved after parameter optimization using Five Adaptive Gradient-based algorithms: ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX. To see how well the Five AGB (Adaptive Gradient-based) algorithms performed on the DDD course data, the average accuracy and recall of each decile/model are computed. The LSTM model with ADAGRAD and ADAMAX optimization produced the best model accuracy and recall values on the DDD course data, with an average model accuracy value of 80%. The results of optimizing the LSTM model parameters for the FFF Course data using the Adaptive Gradient-based algorithms are shown in Table V. Table V displays the accuracy and recall performance that were caused by the LSTM model.

The accuracy and recall values of the LSTM model in the FFF Course significantly improved after parameter optimization using Five Adaptive Gradient-based algorithms: ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX. To see how well the Five Adaptive Gradient-based algorithms performed on the FFF course data, the average accuracy and recall of each decile/model are computed. The LSTM model with ADAGRAD optimization produced the best model accuracy and recall values on the FFF course data, with an average model accuracy value of 85%.

TABLE V
THE LSTM MODEL'S PERFORMANCE AFTER PARAMETER OPTIMIZATION WITH AN ADAPTIVE GRADIENT-BASED ALGORITHM ON THE FFF COURSE

Combining the LSTM Model and AGB (Adaptive Gradient-based) Algorithms					
	Accuracy				
Models	ADAM	NADAM	ADADELTA	ADAGRAD	ADAMAX
0	0.73	0.72	0.74	0.73	0.72
1	0.76	0.75	0.76	0.75	0.76
2	0.79	0.79	0.76	0.79	0.78
3	0.83	0.83	0.79	0.83	0.82
4	0.83	0.84	0.82	0.84	0.84
5	0.85	0.85	0.83	0.86	0.86
6	0.88	0.88	0.87	0.89	0.88
7	0.88	0.87	0.87	0.89	0.87
8	0.88	0.88	0.86	0.88	0.88
9	0.91	0.90	0.90	**0.91**	0.91
10	0.93	0.93	0.93	**0.94**	0.94
Average	0.84	0.84	0.83	**0.85**	0.84

A. Performance Results for the LSTM Model Using the Genetic Optimizations Algorithm

Three LSTM models, LSTM + ADADELTA, LSTM + ADAGRAD, and LSTM + ADAMAX, will be trained and tested at this point by attempting to modify the model hyperparameters. Hyper-parameters are the parameters that are set before training a model and are not learned from the data. They are external to the model and affect its behavior and performance. Hyper-parameters control various aspects of the learning process, such as the model's capacity, regularization, optimization algorithm, and convergence criteria. The selection of appropriate hyper-parameters is crucial as it can greatly impact the model's performance and generalization capabilities. The LSTM-ADADELTA, LSTM-ADAGRAD, and LSTM-ADAMAX models were subjected to hyper-parameter optimization using the Genetic Algorithm. The results of optimizing the LSTM model hyper-parameters for the BBB Course data using the Genetic Algorithms are shown in Table VI. Table VI displays the accuracy and recall performance that were caused by the LSTM model.

TABLE VI
THE LSTM MODEL'S PERFORMANCE AFTER HYPER-PARAMETER OPTIMIZATION WITH GENETIC ALGORITHMS ON THE BBB COURSE

Genetic Optimization Algorithm						
	Accuracy			Recall		
Models	LSTM + GA			LSTM + GA		
	ADADELTA	**ADAGRAD**	**ADAMAX**	**ADADELTA**	**ADAGRAD**	**ADAMAX**
0	0.75	0.75	0.75	0.76	0.76	0.76
1	0.57	0.73	0.75	0.57	0.74	0.76
2	0.60	0.55	0.75	0.60	0.55	0.76
3	0.57	0.66	0.60	0.58	0.67	0.60
4	0.60	0.77	0.75	0.61	0.77	0.76
5	0.74	0.83	0.75	0.74	0.83	0.76
6	0.76	0.85	0.54	0.76	0.86	0.55
7	0.80	**0.88**	0.75	0.81	**0.88**	0.76
8	0.63	0.83	0.75	0.63	0.84	0.76
9	0.85	**0.87**	0.75	0.85	**0.87**	0.76
10	0.77	0.78	0.75	0.77	0.78	0.76
Average	0.69	**0.77**	0.72	0.70	**0.78**	0.73

Based on the results of parameter optimization using ADADELTA, ADAGRAD, and ADAMAX, parameter training and testing on BBB course data using the LSTM model, and hyperparameter optimization using GA, the best model was found in the GA + LSTM + ADAGRAD model. The accuracy values are highest in the 7th and 9th models or deciles, at 87% in the 9th model or decile and 88% in the 7th model or decile, respectively. The GA + LSTM + ADAGRAD model's average recall value is 78%, and its average accuracy value is 77%. The results of optimizing the LSTM model hyper-parameters for the DDD Course data using the Genetic Algorithms are shown in Table VII. Table VII displays the accuracy and recall performance that were caused by the LSTM model.

TABLE VII
THE LSTM MODEL'S PERFORMANCE AFTER HYPER-PARAMETER OPTIMIZATION WITH GENETIC ALGORITHMS ON THE DDD COURSE

Genetic Optimization Algorithm						
	Accuracy			Recall		
Models	LSTM + GA			LSTM + GA		
	ADADELTA	**ADAGRAD**	**ADAMAX**	**ADADELTA**	**ADAGRAD**	**ADAMAX**
0	0.70	0.71	0.68	0.70	0.71	0.69
1	0.74	0.69	0.68	0.74	0.70	0.69
2	0.77	0.76	0.68	0.77	0.76	0.69
3	0.78	0.77	0.68	0.78	0.78	0.69
4	0.79	0.79	0.68	0.80	0.79	0.69
5	0.81	0.76	0.68	0.81	0.77	0.69
6	0.81	0.78	0.68	0.82	0.79	0.69
7	0.83	0.81	0.68	0.84	0.82	0.69
8	0.84	0.84	0.68	0.84	0.84	0.69
9	**0.85**	0.83	0.68	0.86	0.83	0.69
10	**0.86**	0.85	0.68	0.87	0.86	0.69
Average	**0.80**	0.78	0.68	**0.80**	0.79	0.69

Based on the results of parameter optimization using ADADELTA, ADAGRAD, and ADAMAX, parameter training and testing on DDD course data using the LSTM algorithm's subsequent hyperparameter refinement using GA, the best model was found in the GA + LSTM + ADADELTA model. The mean accuracy value of the GA + LSTM + ADADELTA model is 80%, and the mean recall value is 80%. The results of optimizing the LSTM model hyper-parameters for the FFF Course data using the Genetic Algorithms are shown in Table VIII. Table VIII displays the accuracy and recall performance that were caused by the LSTM model.

TABLE VIII
THE LSTM MODEL'S PERFORMANCE AFTER HYPER-PARAMETER OPTIMIZATION WITH GENETIC ALGORITHMS ON THE FFF COURSE

Genetic Optimization Algorithm						
	Accuracy			Recall		
Models	LSTM + GA			LSTM + GA		
	ADADELTA	**ADAGRAD**	**ADAMAX**	**ADADELTA**	**ADAGRAD**	**ADAMAX**
0	0.71	0.72	0.74	0.71	0.72	0.74
1	0.73	0.75	0.75	0.73	0.75	0.75
2	0.80	0.79	0.74	0.80	0.79	0.74
3	0.83	0.79	0.74	0.83	0.79	0.74
4	0.83	0.82	0.74	0.84	0.82	0.74
5	0.86	0.84	0.74	0.87	0.85	0,74
6	0.88	0.86	0.74	0.88	0.86	0.74
7	0.89	0.87	0.74	0.89	0.88	0.74
8	0.87	0.86	0.74	0.87	0.86	0.74
9	**0.91**	0.90	0.74	0.91	0.90	0.74
10	**0.93**	0.92	0.74	0.94	0.92	0.74
Average	**0.84**	0.83	0.74	**0.84**	0.83	0.74

Based on the results of parameter optimization using ADADELTA, ADAGRAD, and ADAMAX, parameter training and testing on FFF course data using the LSTM model's subsequent hyperparameter refinement using GA, the best model was found in the GA + LSTM + ADADELTA model. The mean accuracy worth of the GA + LSTM + ADADELTA model is 84%, and the recall value on average is 84%. In the model or the 10th decile, this FFF model's highest accuracy result is 93%.

The LSTM model has been used to test BBB, DDD, and FFF data; the first test of the LSTM architecture parameter optimization only used Adaptive Gradient-based algorithms; the second test of the LSTM model's parameter optimization also included hyperparameter optimization using a genetic algorithm. The results of LSTM model testing on BBB data after parameter and hyperparameter optimization show that the GA + LSTM + ADAGRAD model is the best. Table IX displays the results of LSTM model testing using BBB data. The LSTM + ADAGRAD model does not undergo hyperparameter optimization, whereas the GA + LSTM + ADAGRAD model does.

TABLE IX
COMPARISON OF LSTM MODELS WITH HYPERPARAMETER OPTIMIZATION AND THOSE WITHOUT HYPERPARAMETER OPTIMIZATION ON BBB COURSE DATA

Comparison of Genetic Algorithms and Adaptive Gradient-based Algorithms for LSTM Model Optimization				
	Accuracy		Recall	
Models	LSTM + ADAGRAD	GA + LSTM + ADAGRAD	LSTM + ADAGRAD	GA + LSTM + ADAGRAD
0	0.75	0.75	0.76	0.76
1	0.58	**0.73**	0.58	**0.74**
2	0.69	0.55	0.70	0.55
3	0.57	**0.66**	0.58	**0.67**
4	0.73	**0.77**	0.73	**0.77**
5	0.78	**0.83**	0.79	**0.83**
6	0.79	**0.85**	0.79	**0.86**
7	0.85	**0.88**	0.85	**0.88**
8	0.64	**0.83**	0.64	**0.84**
9	0.92	0.87	0.92	0.87
10	0.92	0.78	0.93	0.78
Average	0.75	**0.77**	0.75	**0.78**

In light of Table VIII. When LSTM + ADAGRAD and GA + LSTM + ADAGRAD models are compared, the accuracy and recall results of the GA + LSTM + ADAGRAD model, whose hyperparameters have been optimized, significantly improve in Deciles 1, 3, 4, 5, 6, 7, and 8. The obtained accuracy rate was 77% on average. The results of LSTM model testing on DDD data after parameter and hyperparameter optimization show that the GA + LSTM + ADADELTA model is the best. Table X displays the results of LSTM model testing using DDD data. The LSTM + ADADELTA model does not undergo hyperparameter optimization, whereas the GA + LSTM + ADADELTA model does.

TABLE X
COMPARISON OF LSTM MODELS WITH HYPERPARAMETER OPTIMIZATION AND THOSE WITHOUT HYPERPARAMETER OPTIMIZATION ON DDD COURSE DATA

Comparison of Genetic Algorithms and Adaptive Gradient-based Algorithms for LSTM Model Optimization				
	Accuracy		Recall	
Models	LSTM + ADADELTA	GA + LSTM + ADADELTA	LSTM + ADADELTA	GA + LSTM + ADADELTA
0	0.68	**0.70**	0.69	**0.70**
1	0.71	**0.74**	0.71	**0.74**
2	0.72	**0.77**	0.73	**0.77**
3	0.76	**0.78**	0.76	**0.78**
4	0.76	**0.79**	0.76	**0.80**
5	0.75	**0.81**	0.76	**0.81**
6	0.77	**0.81**	0.78	**0.82**
7	0.80	**0.83**	0.80	**0.84**
8	0.81	**0.84**	0.81	**0.84**
9	0.84	**0.85**	0.85	**0.86**
10	0.85	**0.86**	0.86	**0.87**
Average	0.77	**0.80**	0.77	**0.80**

In light of Table X. When LSTM + ADADELTA and GA + LSTM + ADADELTA models are compared, the accuracy and recall results of the GA + LSTM + ADADELTA model, whose hyperparameters have been optimized, significantly improve. The obtained accuracy rate was 80% on average. The results of LSTM model testing on FFF data after parameter and hyperparameter optimization show that the GA + LSTM + ADADELTA model is the best. Table XI displays the results of LSTM model testing using FFF data. The LSTM + ADADELTA model does not undergo hyperparameter optimization, whereas the GA + LSTM + ADADELTA model does.

TABLE XI
COMPARISON OF LSTM MODELS WITH HYPERPARAMETER OPTIMIZATION AND THOSE WITHOUT HYPERPARAMETER OPTIMIZATION ON FFF COURSE DATA

Comparison of Genetic Algorithms and Adaptive Gradient-based Algorithms for LSTM Model Optimization				
	Accuracy		Recall	
Models	LSTM + ADADELTA	GA + LSTM + ADADELTA	LSTM + ADADELTA	GA + LSTM + ADADELTA
0	0.74	0.71	0.74	0.71
1	0.76	0.73	0.76	0.73
2	0.76	**0.80**	0.77	**0.80**
3	0.79	**0.83**	0.79	**0.83**
4	0.82	**0.83**	0.83	**0.84**
5	0.83	**0.86**	0.84	**0.87**
6	0.87	**0.88**	0.87	**0.88**
7	0.87	**0.89**	0.87	**0.89**
8	0.86	**0.87**	0.87	**0.87**
9	0.90	**0.91**	0.91	**0.91**
10	0.93	0.93	0.93	**0.94**
Average	0.83	**0.84**	0.83	**0.84**

In light of Table XI. When LSTM + ADADELTA and GA + LSTM + ADADELTA models are compared, the accuracy and recall results of the GA + LSTM + ADADELTA model, whose hyperparameters have been optimized, significantly improve in Deciles 2, 3, 4, 5, 6, 7, 8, and 9. The obtained accuracy rate was 84% on average.

Using a genetic optimization algorithm, the hyperparameters of three LSTM models with the ADADELTA, ADAGRAD, and ADAMAX algorithms were improved. Based on the analysis and comparison of the LSTM model's performance following hyperparameter optimization on the BBB course data, it was determined that the GA + LSTM + ADAGRAD model was the most effective model. Based on analysis and comparison of the performance of the LSTM model's subsequent hyperparameter refinement on DDD course data, it is demonstrated that the GA + LSTM + ADADELTA model is the best model. After hyperparameter optimization, the GA + LSTM + ADADELTA model also emerged as the top model for the FFF course data.

The expected results of the GA + LSTM + ADAGRAD model for the BBB course with an accuracy of 88% are displayed in Fig. 3. The forecast's outcome of the GA + LSTM + ADADELTA model for the DDD course with an accuracy of 86% are displayed in Fig. 4. The forecast's outcome of the GA + LSTM + ADADELTA model for the FFF course with an accuracy of 92% are displayed in Fig. 5.

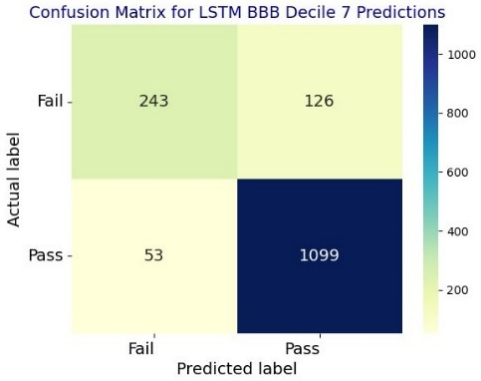

Fig. 3. The BBB course prediction outcomes of the GA + LSTM + ADAGRAD model.

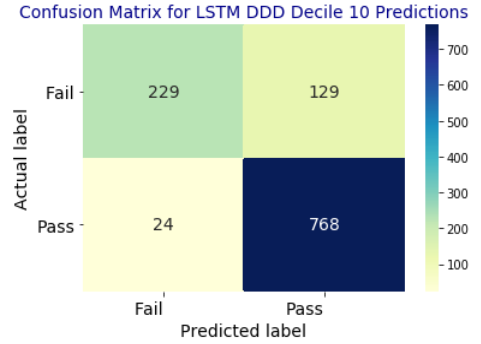

Fig. 4. Prediction outcomes of the GA + LSTM + ADADELTA model on DDD course.

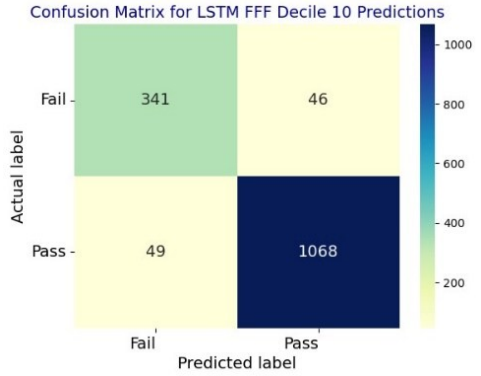

Fig. 5. Prediction outcomes of the GA + LSTM + ADADELTA model on the FFF course.

According to this study's decisions, it can be said that hyperparameters significantly affect how well DL models perform. A similar improvement in performance is brought about by the optimization of model parameters. After optimizing the hyperparameters with a GA algorithm, the accuracy of the LSTM model in this study saw a better improvement.

V. Conclusion

This study aimed to explore two hypotheses regarding the optimization of parameters and hyperparameters in LSTM models for predicting virtual learning environments (VLE). Firstly, the hypothesis suggested that utilizing adaptive gradient-based algorithms, such as ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX, would lead to improved stability, faster convergence, and better handling of gradient issues like vanishing or exploding gradients in LSTM models. Secondly, it hypothesized that employing a genetic optimization algorithm for hyperparameter optimization in LSTM models would result in more optimal and adaptive configurations, considering the complexity and variability of data in VLEs. The results revealed that the adaptive gradient-based algorithms, especially when applied to LSTM models with ADAM, NADAM, ADADELTA, ADAGRAD, and ADAMAX, yielded the most accurate prediction of VLEs. For the BBB course data, the LSTM model with ADADELTA attained the best accuracy of 91%, while for the DDD and FFF course data, the LSTM models with ADAGRAD achieved the best performance with average accuracies of 80% and 85%, respectively. Furthermore, using a genetic algorithm for hyperparameter optimization in LSTM models contributed significantly. The GA + LSTM + ADAGRAD model achieved the highest accuracy of 88% and 87% in the 7th and 9th models, respectively, for the BBB course data, with average recall and accuracy rates of 78% and 77%. Similar trends were observed in the DDD and FFF course data, where the GA + LSTM + ADADELTA model demonstrated the best performance with average accuracies of 80% and 84%, respectively, and achieved the highest accuracy rates of 86% and 93%, respectively. These results shed light on how well adaptive and genetic algorithms work to improve the performance of LSTM models in virtual learning environment prediction. Future research may need to test optimizing LSTM using various metaheuristic algorithms to see how well they perform. Consider incorporating additional features, like data on economic conditions and demographics.

References

[1] M. M. Mashroofa, A. Haleem, N. Nawaz, and M. A. Saldeen, “E-learning adoption for sustainable higher education,” *Heliyon*, vol. 9, no. 6, p. e17505, 2023, https://doi.org/10.1016/j.heliyon.2023.e17505.

[2] Z. Che, C. Peng, and C. Yue, “Optimizing LSTM with multi-strategy improved WOA for robust prediction of high-speed machine tests data,” *Chaos Solitons Fractals*, vol. 178, p. 114394, 2024, https://doi.org/10.1016/j.chaos.2023.114394.

[3] D. G. da Silva and A. A. de Moura Meneses, “Comparing Long Short-Term Memory (LSTM) and bidirectional LSTM deep neural networks for power consumption prediction,” *Energy Reports*, vol. 10, pp. 3315–3334, 2023, https://doi.org/10.1016/j.egyr.2023.09.175.

[4] O. Dermy, A. Roussanaly, and A. Boyer, “Using Behavioral Primitives to Model Students’ Digital Behavior,” *Procedia Comput Sci*, vol. 207, pp. 2444–2453, 2022, https://doi.org/10.1016/j.procs.2022.09.302.

[5] D. Devi and S. Sophia, “GA-CNN: Analyzing student’s cognitive skills with EEG data using a hybrid deep learning approach,” *Biomed Signal Process Control*, vol. 90, p. 105888, 2024, https://doi.org/10.1016/j.bspc.2023.105888.

[6] Q. Fu, Z. Gao, J. Zhou, and Y. Zheng, “CLSA: A novel deep learning model for MOOC dropout prediction,” *Computers & Electrical Engineering*, vol. 94, p. 107315, 2021, https://doi.org/10.1016/j.compeleceng.2021.107315.

[7] S. Gupta, P. Kumar, and R. Tekchandani, “An optimized deep convolutional neural network for adaptive learning using feature fusion in multimodal data,” *Decision Analytics Journal*, vol. 8, p. 100277, 2023, https://doi.org/10.1016/j.dajour.2023.100277.

[8] Y. M. I. Hassan, A. Elkorany, and K. Wassif, “SMFSOP: A semantic-based modelling framework for student outcome prediction,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101728, 2023, https://doi.org/10.1016/j.jksuci.2023.101728.

[9] M. Hlosta, C. Herodotou, T. Papathoma, A. Gillespie, and P. Bergamin, “Predictive learning analytics in online education: A deeper understanding through explaining algorithmic errors,” *Computers and Education: Artificial Intelligence*, vol. 3, p. 100108, 2022, https://doi.org/10.1016/j.caeai.2022.100108.

[10] P. Kumar and A. S. Hati, "Deep convolutional neural network based on adaptive gradient optimizer for fault detection in SCIM," *ISA Trans*, vol. 111, pp. 350–359, 2021, https://doi.org/10.1016/j.isatra.2020.10.052.

[11] B. Li *et al.*, "Adaptive Gradient-Based Optimization Method for Parameter Identification in Power Distribution Network," *International Transactions on Electrical Energy Systems*, vol. 2022, p. 9300522, 2022, https://doi.org/10.1155/2022/9300522.

[12] B. Li *et al.*, "A personalized recommendation framework based on MOOC system integrating deep learning and big data," *Computers and Electrical Engineering*, vol. 106, p. 108571, 2023, https://doi.org/10.1016/j.compeleceng.2022.108571.

[13] Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction," *Expert Syst Appl*, vol. 227, p. 120287, 2023, https://doi.org/10.1016/j.eswa.2023.120287.

[14] Y. Lin, S. Feng, F. Lin, J. Xiahou, and W. Zeng, "Multi-scale reinforced profile for personalized recommendation with deep neural networks in MOOCs," *Appl Soft Comput*, vol. 148, p. 110905, 2023, https://doi.org/10.1016/j.asoc.2023.110905.

[15] J. Martinez-Gil, "Optimizing readability using genetic algorithms," *Knowl Based Syst*, vol. 284, p. 111273, 2024, https://doi.org/10.1016/j.knosys.2023.111273.

[16] A. A. Mubarak, H. Cao, and I. M. Hezam, "Deep analytic model for student dropout prediction in massive open online courses," *Computers & Electrical Engineering*, vol. 93, p. 107271, 2021, https://doi.org/10.1016/j.compeleceng.2021.107271.

[17] M. Neghină, A.-I. Dicoiu, R. Chiș, and A. Florea, "A competitive new multi-objective optimization genetic algorithm based on apparent front ranking," *Eng Appl Artif Intell*, vol. 132, p. 107870, 2024, https://doi.org/10.1016/j.engappai.2024.107870.

[18] K. Niu, Y. Zhou, G. Lu, W. Tai, and K. Zhang, "PMCT: Parallel Multiscale Convolutional Temporal model for MOOC dropout prediction," *Computers and Electrical Engineering*, vol. 112, p. 108989, 2023, https://doi.org/10.1016/j.compeleceng.2023.108989.

[19] L. Peng, T. Zhang, S. Wang, G. Huang, and S. Chen, "Diffusion adagrad minimum kernel risk sensitive mean p-power loss algorithm," *Signal Processing*, vol. 202, p. 108773, 2023, https://doi.org/10.1016/j.sigpro.2022.108773.

[20] Y.-L. Peng and W.-P. Lee, "Practical guidelines for resolving the loss divergence caused by the root-mean-squared propagation optimizer," *Appl Soft Comput*, vol. 153, p. 111335, 2024, https://doi.org/10.1016/j.asoc.2024.111335.

[21] J. Ren and S. Wu, "Prediction of user temporal interactions with online course platforms using deep learning algorithms," *Computers and Education: Artificial Intelligence*, vol. 4, p. 100133, 2023, https://doi.org/10.1016/j.caeai.2023.100133.

[22] H. Waheed, S.-U. Hassan, R. Nawaz, N. R. Aljohani, G. Chen, and D. Gasevic, "Early prediction of learners at risk in self-paced education: A neural network approach," *Expert Syst Appl*, vol. 213, p. 118868, 2023, https://doi.org/10.1016/j.eswa.2022.118868.

[23] S. Sageengrana, S. Selvakumar, and S. Srinivasan, "Optimized RB-RNN: Development of hybrid deep learning for analyzing student's behaviours in online-learning using brain waves and chatbots," *Expert Syst Appl*, vol. 248, p. 123267, 2024, https://doi.org/10.1016/j.eswa.2024.123267.

[24] T. Xu, P. Xu, C. Yang, Z. Li, A. Wang, and W. Guo, "An LSTM-stacked autoencoder multisource response prediction and constraint optimization for scaled expansion tubes," *Appl Soft Comput*, vol. 153, p. 111285, 2024, https://doi.org/10.1016/j.asoc.2024.111285.

[25] K. Wang *et al.*, "A novel GA-LSTM-based prediction method of ship energy usage based on the characteristics analysis of operational data," *Energy*, vol. 282, p. 128910, 2023, https://doi.org/10.1016/j.energy.2023.128910.

[26] M. Uppal *et al.*, "Enhancing accuracy in brain stroke detection: Multi-layer perceptron with Adadelta, RMSProp and AdaMax optimizers," *Front Bioeng Biotechnol*, vol. 11, no. September, pp. 1–15, 2023, https://doi.org/10.3389/fbioe.2023.1257591.

[27] A. K. Silivery, R. M. Rao Kovvur, R. Solleti, L. K. S. Kumar, and B. Madhu, "A model for multi-attack classification to improve intrusion detection performance using deep learning approaches," *Measurement: Sensors*, vol. 30, p. 100924, 2023, https://doi.org/10.1016/j.measen.2023.100924.

[28] Z. Zhao, Y. Bao, T. Gao, and Q. An, "Optimization of GFRP-concrete-steel composite column based on genetic algorithm - artificial neural network," *Applied Ocean Research*, vol. 143, p. 103881, 2024, https://doi.org/10.1016/j.apor.2024.103881.

[29] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput Oper Res*, vol. 152, p. 106131, 2023, https://doi.org/10.1016/j.cor.2022.106131.

[30] Shirdel, M., Di Mauro, M., & Liotta, A. "Worthiness Benchmark: A Novel Concept for Analyzing Binary Classification Evaluation Metrics", Information Sciences, 120882, 2024, https://doi.org/10.1016/j.ins.2024.120882.