Received 2 January 2024, accepted 2 March 2024, date of publication 25 March 2024, date of current version 9 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381619

RESEARCH ARTICLE

An Optimized LSTM-Based Augmented Language Model (FLSTM-ALM) Using Fox Algorithm for Automatic Essay Scoring Prediction

RIDHA HUSSEIN CHASSAB[1,2], LAILATUL QADRI ZAKARIA[1], AND SABRINA TIUN[1]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]Training Department, Staff College, Iraqi Defense University for Military Studies, Baghdad 10001, Iraq

Corresponding author: Lailatul Qadri Zakaria (lailatul.qadri@ukm.edu.my)

This work was supported by Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia.

ABSTRACT The computer-based Automated Essay Scoring (AES) system automatically marks or scores student replies by considering relevant criteria. The methodology systematically categorises writing quality and can increase operational effectiveness in academic and major commercial institutions. To study the projected score, AES relies on extracting numerous aspects from the student's response, including grammatical and textural information. However, the recovered features may result in dimensionality reduction and a challenging-to-understand feature selection procedure. As the number of parameters rises, the model also demands a large cost for processing and training the data. However, these problems worsen the accuracy of score prediction and widen the gap between actual and anticipated results. This study suggested the Fox-optimised Long Short-Term Memory-based Augmented Language Model (FLSTM-ALM) as a solution to these problems for giving successful training to text features; the model uses an augmented learning paradigm. The retrieval score was then analysed and generated using a neural knowledge encoder and retriever. The neural model successfully classifies the output based on this score. The best features are chosen using the Fox optimisation algorithm based on the food-searching category. This choice of parameters solves the exploration and optimisation issues with document classification. The performance of the optimised AES system was assessed using the two datasets, ASAP and ETS, and it demonstrated a high accuracy of 98.92% and a low error rate of 0.096%. Dimensionality reduction can thus be fixed by optimising the FLSTM-ALM model with an appropriate meta-heuristic method, such as the FOX algorithm, which raises the predicted accuracy, recall, and f1 score for the AES model.

INDEX TERMS Automatic essay score, dimensionality reduction, fox algorithm, knowledge encoder, neural knowledge retriever, optimized feature selection process.

I. INTRODUCTION

Artificial intelligence plays a crucial role in education domains, such as question answering [1], [2], [3] and question classification [4]. Automatic Essay Scoring (AES) is among the most challenging in educational institutions. The AES process automatically assigns scores based on student answers [5], [6]. Accurate scoring is only possible through extensive analysis of answer characteristics. However, AES systems face various challenges when analysing student answers [7]. Therefore, the authors focus on AES systems to increase online score performance [8]. The student's answer may contain spelling and grammatical errors, different essay structures, and varying words, sentences, and paragraphs that the AES system should consider as they are subjective [9]. Therefore, these features should be considered to ensure the AES system achieves high accuracy and reduces the difference between the actual and predicted values [9].

Some AES systems produce unsatisfactory results due to the unsuitable assessment of student answers. Hence, the AES training system is advanced by considering the hand-score manner. The training set considers text surface features,

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

subordinate clues, lower case letters, word count, sentence count, and upper case letters [10]. These features are used to construct a mathematical model using different machine-learning techniques. The extracted features are analysed by using various regression linear techniques to support the AES model in achieving a score value similar to the hand-made scoring process [7], [8], [10].

AES models can be divided into supervised and unsupervised learning processes [7], [13], [14]. The training patterns are extracted from the features of supervised learning to analyse the input essay and assign score values. In unsupervised learning, score values are generated for the input essay without learning patterns. Deep learning (DL) approaches are widely used in AES due to their ability to analyse input text by generating learning patterns from parameters. Currently, there are various sub-neural models in the DL approach, such as Recurrent Neural Networks (RNN), Convolution Neural Networks (CNN), Short-Term Long Memory (LSTM), Generative Adversarial Networks (GANs), and Radial Basis Neural Networks (RBN) [15], [16]. These DL models use various learning regression techniques to predict the score value. However, DL models face various challenges, such as dimensionality and redundancy issues, which cause optimisation problems and impact classification rates. The issues with prediction error and optimisation negatively impact the overall performance of score prediction.

Therefore, this study focuses on improving score prediction and decreasing the error rate. The discussed issues are resolved by applying the pre-training model with a few parameters and a fine-tuning process to achieve this. The training process employs the concept of a knowledge retriever to enhance the overall system performance. The knowledge retriever examines a set of documents and retrieves answers to each question. Then, the knowledge-augmented encoder uses the knowledge retrieved information to select the right answer. The pre-training process involves using the masked language model (MLM). After that, fine-tuning is carried out using open QA. Then, long-short-time memory neural networks (LSTM) are used to compute the score value for the given input and solve complex computation problems.

Additionally, the Fox optimisation algorithm is used to address convergence issues and improve efficiency. Our proposed model, or Fox-optimised LSTM-based augmented language model (FLSTM-ALM), is used to solve optimisation problems during the updating of network parameters and predicts the optimal solution in the searching space. Eventually, the optimisation enhances the overall prediction accuracy and reduces output variations in the AES system. Afterwards, the performance of the enhanced AES system was evaluated based on two datasets using the accuracy, F1-score, and Quadratic Weighted Kappa (QWK) metrics.

This paper is organised as follows, Section II shows the related work of recent research, Section III describes the Fox optimised LTSM-based augmented language model (FLSTM-ALM), Section IV shows the results.

II. RELATED WORK

This section examines and analyses the different techniques used in the AES system. The main objective of this analysis is to increase understanding of the AES score prediction.

In [5], the authors attempted to increase the AES system by extracting the lexical vectors of the text from various encoding methods. Then, they proposed a model based on the stacking method. They also conducted a comparative analysis of different benchmark datasets for English essay scoring on Kaggle. The results obtained using the stacking method achieved an improvement of 1.0% and 2.8% compared to the baseline models of neural networks and feature-engineered models.

A study [8], the authors attempted to improve the AES system by using a tree-structured Parzen Estimator-based neural model. This research derived features from the BERT structure and utilised the XGboosting technique to process the extracted features from the BERT model to create a pre-training model. As a result, the pre-training model was able to predict the input essay score value with 0.829% accuracy using the Ukara dataset.

In [13], the authors compared embedding models such as Glove, Elmo, and Google Sentence Encoder (GSE) using similarity methods such as Jaccard and cosine. The ASAP benchmark dataset was used to evaluate the models. The cosine similarity method was used to find the similarity between the student's essay input and the model essay. GSE had the highest score value.

Another study [17] created an AES system that applies a co-attention deep learning (CA-DL) architecture. The system uses a Glove pre-training model to drive the embedding vectors to the convolution layers. The LSTM neural network is used to analyse the results by generating a sentence embedding vector for the input essay. A co-attention layer is applied to identify a similar sentence between the model and the student's answer. The final CNN layer was used to score the input essay. The system's accuracy was evaluated using the ASAP dataset and achieved 81.5%.

In a study [18] employed multi-task learning (MTL) and a deep learning model (DL) to score the AES system. The authors analysed the structure of the essay to extract features such as vocabulary size, word count, essay vocabulary, and organisation. They used the convolution layer for the Glove pre-training model to derive the word embedding details. Then, the LSTM network was applied to process the obtained embedding features to predict the score value for the input essay. Finally, the system achieves 76.4% accuracy.

In [19], the authors developed an AES system using two models, LSTM and CNN. They first used a Glove pre-training model to generate word embedding vectors from the input essay. These embedding vectors were then given as input to the LSTM network to predict the score value. The system was evaluated using the ASAP benchmark dataset and achieved 72.65% accuracy.

In a study [20] aimed at developing an AES to reduce the difficulties in manual essay scoring. However, the AES system should consider prompt content, cohesion, development of ideas, and coherence. These parameters are difficult to address using existing AES techniques.

In [21], the researchers were interested in developing an AES system by combining 30 manually extracted features, 300 word2vec representations, and 768 word embedding features using the BERT model. The results showed an accuracy of 77.2 ± 1.7 Kappa statistics for the rescaled regression problem and 75.2 ± 1.0 for the quantised classification problem, using a benchmark dataset of about 12,000 essays divided into eight groups. The results provide directions for researchers in the field to use manually extracted features with deep-encoded features to develop a more reliable AES model.

In a study [22], the authors attempted to incorporate content and structural analysis to provide a complete grading system. They recognise that revision and feedback are essential aspects of the writing process. Thus, the model was built using the concepts of LSTM and entity detection and a user interface was incorporated to input an essay and obtain its score, along with a breakdown analysis of the essay.

Table 1 shows the comparison of the reviewed related work. These related works use various neural models and embedding techniques. However, current AES systems fail to solve the optimisation problem and improve the prediction rate to minimise the difference between predicted score values and actual values.

The significance of current research can aid in developing more accurate and efficient models for AES tasks by tackling the problems of dimensionality and redundancy and utilising knowledge retrieval and knowledge-augmented encoder techniques. The study can help to further increase the accuracy of natural language processing systems by optimising the network parameters with the Fox algorithm and using the LSTM technique to forecast essay results. For tasks like automated grading, sentiment analysis, and information retrieval, the suggested approach may impact daily life [23].

Thus, the discussed issues can be solvedby applying the pre-training model with a few fine-tuning processes for the parameters. Then, the knowledge retriever is used to analyse a set of documents. The knowledge-augmented encoder is then used to retrieve the correct answer. Finally, a suitable meta-heuristic optimisation algorithm can be applied to select the optimal parameters while updating network parameters and predicting the best solution in the search space.

Therefore, the main focus of our study is to: (i) minimise the dimensionality and redundancy issues by applying a pre-training model with a few parameters for the fine-tuning process, (ii) improve the overall system performance by using the concept of a knowledge retriever to analyse a set of documents and a knowledge augmented encoder to retrieve the correct answer, (iii) minimise the prediction error rate by selecting the optimum network parameters in the search space of the meta-heuristic (Fox algorithm) algorithm, and (iv) maximise the essay score prediction accuracy by using the LSTM approach.

TABLE 1. Related works.

Ref. no	Implemented Algorithm	Techniques	Dataset	Evaluation metric with the result
[5]	Stacked model	Lexical vector extraction of text	ASAP dataset	1.0%~2.8% higher essay score
[8]	Tree-structured Parzen Estimator-based neural model	BERT, XGBoost	Ukara dataset	82.9 accuracy
[13]	Glove, Elmo, and Google Sentence Encoder (GCE) models	Jaccard and cosine similarity	ASAP dataset	GCE outperform other models
[17]	CA-DL	Glove pre-training model, The LSTM neural network	ASAP dataset	81.5 accuracy
[18]	MTL-DL	Glove pre-training model, LSTM neural network	ASAP dataset	76.4 accuracy
[19]	LSTM+CNN	Glove pre-training model, LSTM neural network	ASAP dataset	72.65% of accuracy.
[21]	Rescaled regression (RR) & Quantised classification (QC)	Manual features, Word2Vec& BERT	ASAP dataset	77.2 ±1.7 for RR 75.2 ± 1.0 for QC

III. FOX OPTIMISED LSTM-BASED AUGMENTED LANGUAGE MODEL (FLSTM-ALM)

This research proposes a fox-optimised LSTM-based augmented language model for predicting the score value for input essays. The main objective of this research is to maximise accuracy by minimising the classification rate and solving optimisation problems. The current pre-training models extract various features from input essay scores, leading to dimensionality reduction and optimised feature selection processes [24]. However, optimisation problems and prediction errors affect the overall system performance. The FLSTM-ALM model is applied to overcome the research problem.

The novelty of this work resides in its creative suggestion of a method for resolving AES systems' problems and enhancing their performance. The Fox optimisation technique is used for feature selection and optimisation, which solves the dimensionality reduction problem by pre-training the model and gives the AES system great accuracy using LSTM and a low error rate. Existing AES systems cannot yet solve the optimisation problem, nor can the prediction rate be increased to reduce the discrepancy between predicted and actual score values.

In the LSTM architecture provided, we have configured several essential parameters to create an effective model for sequence classification tasks. The first crucial parameter is the number of LSTM layers, which determines the depth of our model. Multiple layers can capture increasingly abstract features from the input sequences, and here we have included two LSTM layers. The 'return_sequences=True' setting for the first LSTM layer ensures that it passes sequence data to subsequent layers. The learning rate, another critical factor, has been set at 0.001, regulating the step size during optimisation with the Adam optimiser. The learning rate is a vital hyperparameter that influences training convergence. Finally, we have defined the batch size as 32, which impacts how many sequences are processed together during each training step. Batch size choices depend on available computational resources and dataset size. These parameters should be fine-tuned and customised according to the specific task and dataset characteristics to achieve optimal model performance [25].

Recursive Ensemble Learning Model (RELM) architecture emerges as an intriguing divergence from conventional deep learning paradigms, such as LSTM, offering a distinct approach for addressing the intricate confluence of temporal dependency modelling and mitigating overfitting. In the quest to augment our research endeavours, we have discerned the potential of an adapted hybrid architecture. This adaptation entails the synergistic amalgamation of RELM's ensemble-based methodology and LSTM's sequential modelling prowess. The incorporation of this adapted architecture into our research framework serves as an avant-garde venture, seeking to exploit the complementary attributes of these models, thereby pursuing more refined predictive capabilities while navigating the intricacies inherent to our specific domain. This innovative amalgamation holds promise for advancing our scholarly undertakings and further enriching the contemporary discourse surrounding machine learning methodologies, ultimately contributing to the development of more robust and computationally efficient models [26].

First, the classification problem is resolved within the pre-training model itself. The trained patterns help identify similarities between the new and trained text features. If the computed similarity values have a low deviation, the essay attains a maximum score; the score values are allocated

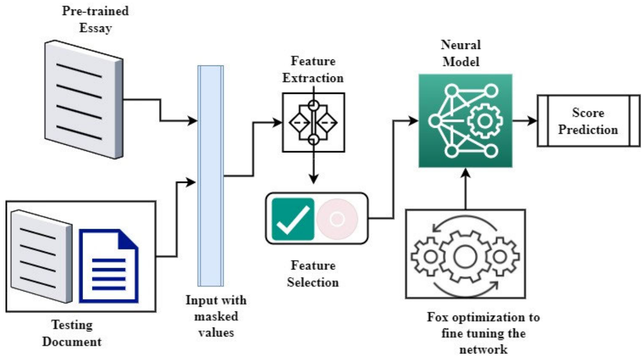


FIGURE 1. Fox optimised neural model-based score prediction of AES system.

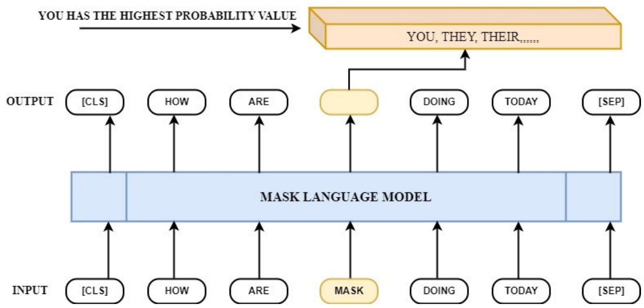


FIGURE 2. Representation of MLM.

depending on the percentage of similarity values. Different language models, such as T5, Roberta, and BERT, process huge amounts of data and perform numerous NLP tasks. These processed tasks are stored in the dataset and used for analysing the essay in different aspects. It also addresses predicted redundancy issues. The redundancy and dimensionality issues affect the overall system performance and make it difficult to interpret the model results.

Furthermore, the model requires a high cost while processing and training the data because the number of parameters increases gradually. These issues are addressed by applying a pre-training model with a few fine-tuning processes. The training process uses the concept of knowledge retrieval to improve the overall system performance. Knowledge retrieval can analyse a set of documents and retrieve answers to questions. Then, a knowledge-augmented encoder is utilised to select the correct answer. The pre-training process involves using a masked language model (MLM). After this, a fine-tuning process is performed using Open QA. The FLSTM-ALM process is illustrated in Figure 1.

A. PHASE 1: INPUT WITH MASK LANGUAGE MODELING (MLM)

Mask Language Modelling (MLM) is a self-supervised learning model that aims to create a pre-training set utilised in natural language processing (NLP) models. In this process, input samples or tokens are replaced with mask values, and then the created model identifies and retrieves the relevant content to fill the space. The MLM process is illustrated in Figure 2.

This research uses augmented language modelling to perform the mask process on the given input sample. The augmented model processes the inputs using two steps: retrieval and prediction. It considers the input sample x, which is analysed in Corpus Z for retrieving the document z. The analysis computed from the study of the distribution z is defined as $p(z|x)$. The retrieved content z is analysed according to the input x, and the output y is predicted. The output computation is defined as $p(y|z,x)$. During these computations, y is generated by using the likelihood computation from the latent variable z, which is defined using eqn (1).

$$p(y|x) = \sum_{z \in Z} (y|z,x)\, p(z|x) \tag{1}$$

The masked values-related documents are retrieved according to the knowledge retriever performed.

$$p(z|x) = \frac{\exp f(x,z)}{\sum_{z'} \exp f(x,z')} \tag{2}$$

In eqn (2), the $f(x,z)$ is estimated from the embed input and embedded document, which is defined as $f(x,z) = Embed_{input}(x)^T\, Embed_{doc}(z)$. The embedding input and documents functions help to map x and z in the dimensional vector d. Then the relevance score $f(x,z)$ is computed between x and z by performing the vector embedding's inner products. Finally, the softmax function is applied to the relevance score to calculate the retrieval distribution of the word. In addition to this, prefixed [CLS] and [SEP] tokens are utilised to perform the span joining process. The span joining is done by applying eqn (3a and 3b).

$$Join_{BERT}(x) = [CLS]x[SEP] \tag{3a}$$

$$Join_{BERT}(x_1, x_2) = [CLS]x_1[SEP]x_2[SEP] \tag{3b}$$

Then the final token is obtained with the help of the vector product, which is done by the transformer by applying eqn (4a and 4b).

$$Embed_{input}(x) = W_{input} BERT_{CLS}(Join_{BERT}(x)) \tag{4a}$$

$$Embed_{doc}(z) = W_{doc} BERT_{CLS}\left(Join_{BERT}\left(z_{title}, z_{body}\right)\right) \tag{4b}$$

After performing the knowledge-retrieving process, the e-augmented encoder process is performed. The knowledge retrieval process is defined in eqn (5a and 5b).

$$p(y|z,x) = \prod_{j=1}^{J_x} p(y_j|z,x) \tag{5a}$$

$$p(y_j|z,x) \propto exp\left(w_j^T BERT_{mask(j)} Join_{BERT}\left(x, z_{body}\right)\right) \tag{5b}$$

In the above eqn (5a) and (5b), transformer vector output is defined as the $BERT_{mask(j)}$ that is related to the jth masked token. For x, the total number of masked values is represented as Jx, and the learned word embedding is denoted as w_j.

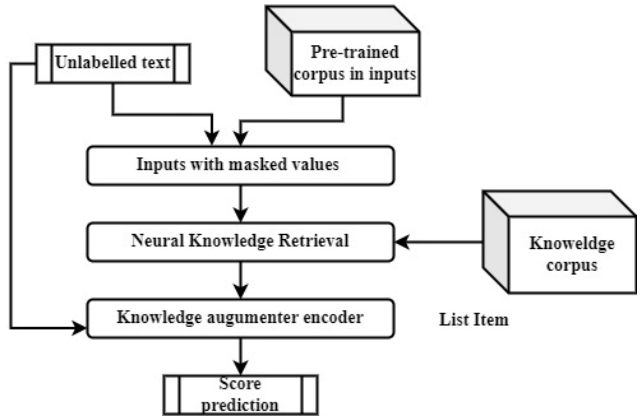


FIGURE 3. Augmented model-based learning.

The mask augmentation model representation is illustrated in Figure 3.

The fine-tuning process aims to improve the automatic score prediction process. The fine-tuning process is performed as follows:

$$p(y|z,x) \propto \sum_{s \in S(z,y)} \exp(LSTM\left(h_{START(s)}; h_{END(s)}\right)) \tag{6}$$

In eqn (6), transformer vector outputs are related to the start $BERT_{START(s)}$ and end $BERT_{END(s)}$ token of span s.

B. PHASE 2: EXTRACT AND SELECT THE FEATURES BY USING LSTM

In this phase, the LSTM is used to compute the score value for the given input x. During document analysis, the main challenge is computation. The document analysis calculates the marginal probability distribution value when document summations are estimated, and this computation creates difficulties in the entire score prediction process. The problem is overcome by computing the embedded input and the document's maximum inner product search value. This calculation is done for a larger number of iterations that help to predict the highest relevance score value. According to the score value, the top documents are expected to be successful.

The retriever learning process helps predict the score value during the testing process. For every document z, the retriever identifies the score value using a gradient function, and the retriever alters the score value $f(x,z)$ by $r(z)$. The retriever change value is positive when it increases and negative when it decreases. The $r(z)$ multiplier value is positive when $p(y|z,x) > p(y|x)$. The $p(y|z,x)$ is used to predict the correct output while analysing document z. During the computation, probability values are estimated while computing the output y. The $p(y|x)$ is used to identify the expected output value from the random sample documents $p(z|x)$. Then the retriever change score $r(z)$ is computed by applying eqn (7a and 7b) as follows:

$$\nabla \log p(y|x) = \sum_{z \in Z} r(z) \nabla f(x,z) \tag{7a}$$

$$r(z) = \left[\frac{p(y \mid z, x)}{p(y \mid x)} - 1\right] p(z \mid x) \tag{7b}$$

The predicted encoder and retriever learning values are processed using an optimised approach with LSTM. LSTM is an effective recurrent neural network (RNN) that can process a sequence of information. The method solves the vanishing gradient problem while analysing sequences of text. The recurrent network uses persistent memory to analyse the computed embedded text and documents. The main reason for choosing this network is that it stores previously processed information, reducing unwanted feature processing time. In addition, the network eliminates the long-term dependency problem.

The network has three gates: the forget gate, the input gate, and the output gate. The incoming inputs are processed to predict whether they come from the current time step. The second part examines new inputs from the input cell, and the third part updates the current timestep output. The network has a hidden state with a previous timestamp $H_{(t-1)}$ and it has a cell state with present and previous timestamps that are denoted as $C_{(t-1)}$ and $C_{(t)}$. The neural network cell state is denoted as the long-term memory, and the hidden state is represented as the short-term memory. These memory models can effectively process time-series data. The network has feedback connections, and the memory is used to save the current and previously analysed data until the next iterations are performed.

For each iteration, the LSTM network effectively updates previously processed information. Frequent iteration minimises classification or prediction errors while producing the final output value. The error rates are reduced by updating or fine-tuning the network parameters. The computation process uses a high learning rate to improve the overall score prediction performance. The fine-tuning process minimises the error rate and solves optimisation problems and convergence issues. The updating procedure also considers the learning rate, weight, and bias of the updating process. Generally, the LSTM process uses a gradient descent optimisation method for updating the network parameters. The working process of the LSTM network is illustrated in Figure 4.

The LSTM network uses inputs processed through three gates to obtain an output, which is used to predict essay scores. The output is computed using activation function inputs, the sum of weighted values, and bias to produce the output. The LSTM uses the tanh and sigmoid activation functions to predict the output value. Then the sigmoid function of the LSTM network is defined using eqn (8).

$$\sigma(x) = (1+e^{-x})^{-1} \tag{8}$$

Here, x is denoted as input, and the output is obtained as 1 or 0, determining whether the written content matches the pre-trained model content. If the network returns 0, the contents are dissimilar. Otherwise, they are similar.

During this computation, the Fox optimisation algorithm updates the network parameter to minimise the difference between actual and predicted values. The number of neurons in each layer, the rate of learning, and other hyperparameters were probably tuned by tuning the parameters using the Fox optimisation method utilised for LSTM configuration to find the ideal settings for the number of hidden layers. The suggested method involves several components, including optimising the LSTM model, using a neural knowledge encoder and retriever to examine a collection of documents and extract pertinent information, and selecting the best features using the Fox optimisation algorithm. Although the LSTM model's optimised parameters are not specifically described in the research material, it is likely that the proposed work fully relies on the Fox optimisation technique implemented to choose the parameters to increase prediction accuracy.

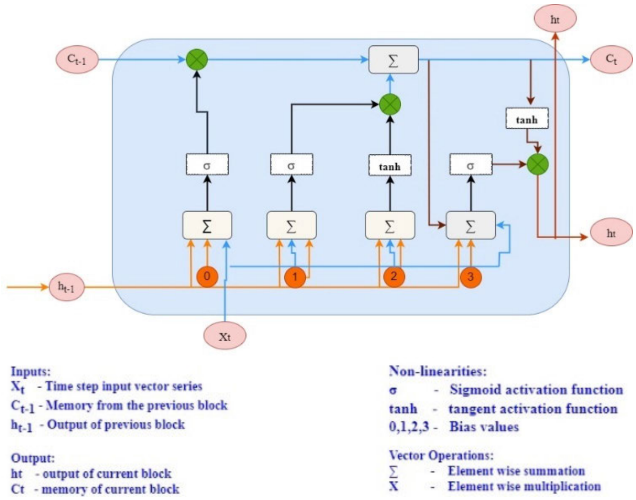


FIGURE 4. Structure of LSTM networks.

Based on the similarity, scores are allocated to the new document. In addition, the network uses the tangent function as the default function, which is estimated using eqn (9), with a value ranging from -1 to 1.

$$\left.\begin{aligned} \tanh(x) &= \frac{\sin x}{\cos x} = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \\ \sin(x) &= \frac{e^{iz}-e^{-iz}}{2i} \\ \cos(x) &= \frac{e^{iz}-e^{-iz}}{2} \end{aligned}\right\} \tag{9}$$

Therefore, the tanh activation is utilised in the output computation to get the output value. In eqn (9) (e^{-z}, e^{-iz}) is represented as the predicted value related to the exponential value, and (e^{z}, e^{iz}) is represented as the actual value related to the exponential value. As stated, the LSTM network has a memory state of the current time $C_{(t)}$ and previous time steps $C_{(t-1)}$. The layers include or delete information from the cell state for every input processing or iteration. In addition, the cell state controls the forget, input, and output gates. The forget gate determines the input and removes the cell state, which is done with the help of the activation layer (eqn 10).

$$f_t = \sigma_g(W_f \circ [h_{t-1}, x_t] + b_f) \tag{10}$$

In eqn (10), element-wise multiplication is denoted by $\circ$, and the input value at the time step t is denoted as x_t, and the sigmoid activation function is represented as σ, computed from eqn (8). Then t and t-1 time step hidden states are denoted as ht and h_{t-1}, forget gate is represented as f, the cell candidate is denoted as g, forget gate weight values are represented as W_f and the bias values are represented as b_f. If the cell decides the input should be included in the state, the activation function is applied according to eqn (11).

$$i_t = \sigma_g(W_i{}^\circ[h_{t-1}, x_t]+b_i) \tag{11}$$

In eqn (11), the network uses the sigmoid activation function to compute the output value. The input gate at the t time step is defined as i_t which uses the time step t input value x_t, input gate bias value b_i, weight value W_i and t-1 time step hidden state value h_{t-1}. The tanh activation function generates the present candidate vector $\hat{C}_t$ utilised for the updating process.

$$\hat{C}_t= \tanh(W_c{}^\circ[h_{t-1}, x_t]+b_c \tag{12}$$

In eqn (12), tanh is represented as the tangent activation derived from eqn (9). Then the previous cell state c_{t-1} is updated as the current cell c_t. After that, the forgotten value is the product with the current cell state, input gate, and current candidate vector, and this computation is derived in eqn (13).

$$c_t = f_t{}^\circ c_{t-1} + i_t{}^\circ \hat{C}_t, \tag{13}$$

Finally, the output is computed with the help of the cell state. First, the sigmoid function is applied to the last cell value. Then the output is transferred as the input to the next activation function, and the computation is defined using (14).

$$\left.\begin{aligned} o_t &= \sigma_g\left(W_o^\circ\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t &= o_t^\circ tanh\left(c_t\right), \end{aligned}\right\} \tag{14}$$

In eqn (14), the previous state activation function process is defined as h_{t-1} and the time step t-related output gate is represented as o_t. The output is predicted based on the above computation. Scores are allocated to the student based on the similarity values.

C. PHASE 3: FOX OPTIMIZATION TO FINE-TUNE THE NETWORK

In this research work, the Fox optimisation (Fu et al., 2022) algorithm is used to update the network parameter to minimise the difference between actual and predicted values by minimizing the deviation and optimisation problems. The algorithm can solve convergence issues,attain high efficiency, and resolve optimisation problems while updating network parameters. It predicts the best solution in the search space. The algorithm effectively minimises optimisation problems compared to other methods. The Fox algorithm intends to choose the learning rate and update the network parameters by choosing the appropriate weight. The selected weight values reduce the loss value and the difference between the actual and predicted output.

The Fox optimisation algorithm works according to the inspiration of the flying fox's survival strategies. The optimisation algorithm addresses the discrete and continuous optimisation problems while updating the network parameter. The algorithm uses fuzzy logic to analyse individual parameters, effectively predicting solutions to problems. The fox's food searching process is utilised to select the optimal solution. The algorithm has two phases; in the first phase, territory exploration searching is performed. The first phase is called global search because the fox computes the distance between the fox and its prey. In the second phase, the fox zeroes in on prey by moving, it is called local search.

Initialisation. The search space has n number of foxes, and it is denoted as the $\bar{x} = \{x_1, x_2, \ldots\ldots x_n\}$. Each fox in n coordinates has the t number of iterations for searching for the optimal food. Then the notation of specific fox at t iteration is denoted as $\left(\bar{x}_j^i\right)^t$. The fox moved in the search space, finding the optimal solution according to the global and local searches. Considered Fox x_i it has a specific function $f \in \mathbb{R}^n$ that is used to find the optimal solution. The function $f\left(\bar{x}_j^i\right)^t$ select the food source if the computed global search has a minimum value at the point (a,b); here, $a,b \in \mathbb{R}$.

Global Searching for Food. After initialising the fox in the search space, the food-searching process, represented as global searching, is performed. Each fox plays a crucial role in the herd, and the food-searching process decides the fox family members. The foxes search for food in local areas and other territories of exploration. If the territory has no food, the fox moves to another location. The exploration information is shared by the foxes, which helps determine their survival rate. The exploration value is computed by the fitness value, and only a few foxes can compute the value, which is transmitted to the other family members.

The foxes are arranged according to the fitness value to achieve this goal, and here, the Euclidean distance measure is estimated to predict the best value, which is computed using eqn (15).

$$d\left(\left(\bar{x}^i\right)^t, \left(\bar{x}^{best}\right)^t\right) = \sqrt{\left\|\left(\bar{x}^i\right)^t - \left(\bar{x}^{best}\right)^t\right\|} \tag{15}$$

This search process moved toward each individual to compute the best solution, and it has been defined using eqn (16).

$$\left(\bar{x}^i\right)^t = \left(\bar{x}^i\right)^t + \alpha\ sign\left(\left(\bar{x}^{best}\right)^t - \left(\bar{x}^i\right)\right) \tag{16}$$

In eqn (16), randomly selected scaling parameters are denoted as α; $\alpha \in (0, d\left(\left(\bar{x}^i\right)^t, \left(\bar{x}^{best}\right)^t\right)$. The α value is selected at every iteration. During the search process, if fox's new position is better than the existing one, it will be replaced with the new one; otherwise, it will move towards the search. According to the results, the remaining family members move towards the explored areas, and foxes hunt for food when they find it in their search. This random movement and exploration process gives the best solution to the global search process.

Local search for local habitat while traveling. The fox searches for food in its territory and slowly moves closer to its prey when it is not noticed. Then fox creates a situation in which it believes the prey will not be affected and tries to attack it by surprise. For every attack, the model randomly generates a value denoted as $\mu \in (0, 1)$. The μ value only determines whether the prey is nearer to the Fox, as defined by using (17).

$$\begin{cases} move\ closer\ if\ \mu > 0.75 \\ stay\ and\ disguise\ if\ \mu \leq 0.75 \end{cases} \tag{17}$$

In eqn (17), μ determines the movement of the fox in the search space. The movement radius is determined using two parameters, such as the scaling parameter ($a \in (0, 0.2)$) and observation angle ($\emptyset_0 \in (0, 2\pi)$). The value of a is set once because the distance of each fox with population changes while searching for food. In addition, the vision radius is also computed with the help of the $\emptyset_0$.

$$r = \begin{cases} a\dfrac{\sin(\emptyset_0)}{\emptyset_0} if\ \emptyset_0 \neq 0 \\ \theta \qquad if\ \emptyset_0 = 0 \end{cases} \tag{18}$$

In eqn (18), θ is used to identify the weather condition in the search space, and it has a value between 0 and 1. Therefore, the fox moves in the search space according to the spatial coordinates defined in eqn (19).

$$\begin{cases} x_0^{new} = ar \cdot cos\,(\emptyset_1) + x_0^{actual} \\ x_1^{new} = ar.\sin\,(\emptyset_1)\ \text{ar} \cdot cos\,(\emptyset_2) + x_1^{actual} \\ x_{n-1}^{new} = ar \cdot \sin\,(\emptyset_1)\ ar.\sin\,(\emptyset_2) + \cdots + arsin\,(\emptyset_{n-1}) \\ +x_{n-1}^{actual} \end{cases} \tag{19}$$

In eqn (19), angular values are randomised, and they have a value between ($\emptyset_0 \in (0, 2\pi)$). This process is performed until the fox finds its next prey in the search space.

Reproduction and herd leaving. After finding the optimal food sources, the fox must leave the herd and consider reproduction factors. Foxes face several issues due to human hunting and a lack of food resources. These are the two main reasons foxes move from one location to another. From the collection of foxes, 5% of foxes with the worst fitness according to the fitness function move towards reproduction or migration. Therefore, the existing values should be replaced by new offspring values. Consider a search space with two individuals, $\left(\overline{x}^{(1)}\right)^t$ and $\left(\overline{x}^{(2)}\right)^t$ defined as the alpha couple. The habitat center value is estimated using these best values using eqn (20).

$$\left(habitat^{(center)}\right)^t = \frac{\left(\overline{Tx}^{(1)}\right)^t + \left(\overline{x}^{(2)}\right)^t}{2} \tag{20}$$

The Euclidean distance of habitat center value is computed for the alpha couple by using eqn (21).

$$\left(habitat^{(diameter)}\right)^t = \sqrt{\left\| \left(\overline{x}^{(1)}\right)^t - \left(\overline{x}^{(2)}\right)^t \right\|} \tag{21}$$

After computing the habitat diameter and center, the replacement is determined according to the random parameter$k \in (0, 1)$.

$$\begin{cases} new\ individual\ if\ k \geq 0.45 \\ reproduction\ of\ alpha\ couple\ if\ k < 0.45 \end{cases} \tag{22}$$

In eqn (22), the foxes have moved away from their family and search for food in a new search space. In the second case, the reproduced values replaced the existing ones, making the food-searching process more efficient. From the computation, a new individual is generated according to eqn (23)

$$\left(\overline{x}^{(reproduced)}\right)^t = k\frac{\left(\overline{x}^{(1)}\right)^t + \left(\overline{x}^{(2)}\right)^t}{2} \tag{23}$$

The fitness value (eqn 24) is estimated during the computation by analysing the number of individuals in the search space.

$$fit = \frac{\sum_{k=0}^{individual} \left| f\left(x^k\right) - f(x^{ideal}) \right|}{individuals} \tag{24}$$

In eqn (24), the position of k individuals is denoted as x^k and the analytical solutions of individuals are denoted as x^{ideal}. The maximum and minimum functions is represented as $f(.)$. According to the above process, the weight parameters are selected from the computed values. The search process is performed globally and locally, which helps improve the overall optimised solution selection process. This process successfully minimises classification problems and convergence issues. The neural model returns new testing documents, like the pre-trained model. If it has a high similarity, it returns 1, and the score is allocated accordingly, otherwise returns 0. Then, the effectiveness of the system is evaluated using the results.

IV. RESULT IN ANALYSIS AND DISCUSSION

A. RESULT ANALYSIS

In our research, we have adopted an innovative approach by incorporating the Fox optimisation algorithm, a recent advancement in optimisation methods, to address the intrinsic complexity associated with LSTM-based models. LSTM architectures, renowned for their capacity to capture long-term dependencies in sequential data, are often challenged by computational complexity, particularly in high-dimensional datasets such as textual sequences.

The decision to employ the Fox optimisation algorithm is grounded in the need to provide an effective solution to this complexity issue. The Fox algorithm, characterised by its data-driven and intelligent feature selection capabilities, serves to alleviate the dimensionality problem by retaining the most pertinent features, thus enhancing computational efficiency and model performance.

Our work stands out by virtue of its novel integration of the Fox optimisation algorithm with LSTM, offering a unique approach that differs from conventional optimisation algorithms typically employed in the field. While we acknowledge the existence of established optimisation methods, we contend that this cutting-edge amalgamation demonstrates significant promise in addressing the complexity challenge associated with LSTM models. Through our research, we have substantiated that the Fox optimisation algorithm not only enhances predictive accuracy but also optimises feature selection, ultimately contributing to the advancement of the field.

This section first analyses the efficiency of extracting features to update the parameters during the fine-tuning by applying the Fox-optimised LSTM-based augmented language model (FLSTM-ALM) based AES system. In the second section, we evaluated the proposed optimised AES system, the FLSTM-ALM, against other deep-learning AES systems.

(i)Feature Extraction.

As discussed, a set of prompts is collected from the database, which is processed by applying preprocessing techniques and feature engineering procedures. The masked language model analyses the extracted features, which mask a few sample portions. Then, the pre-training process is performed to predict the learning pattern. The augmented model is applied to analyse the inputs, using a neural knowledge retriever and a knowledge augment encoder. The successful utilisation of these modules in mask language modelling helps to identify the learning patterns created according to the count-based features, morphological features, POS tagging, and lemma features.

The extracted and optimised features are selected according to the fox-optimisation algorithm, which uses global and local searches to select the best features from the set. Then, the LSTM neural model is applied to the augmented model to classify the output. The network parameters' learning rates are fine-tuned to improve the overall essay score prediction efficiency. The network employs different epochs, learning rates, activation functions, batch normalisation, and gradient descent parameters to enhance the data processing. The effective utilisation of objective function, fitness values, local search, and global search helps to select the optimal feature and update the network parameters effectively.

The introduced approach can effectively solve optimisation, dimensionality, and classification problems. Based on the optimisation algorithm. The fox optimisation algorithm chooses the more appropriate features from the extracted feature list. The algorithm uses a local and global search process to effectively compute the distance values, closer values, positions, and angular values. These parameters help predict the more relevant features and reproduce and select features effectively. After selecting the features, augmented language modelling, transformer, embedding inputs, and encoding documents are useful in predicting the score value.

(ii) FLSTM-ALM against other deep learning AES systems.

The prediction process is performed according to the pre-trained model. Therefore, the system's efficiency is evaluated using both training and testing models. The system's effectiveness is determined using accuracy, F1-score, and QWK parameters. The obtained results are evaluated for both the training and testing sets, as shown in Table 2.

TABLE 2. Performance analysis of FLSTM-ALM.

Training

Model	Accuracy	F1-Score	QWK
CA-DL	81.49	85.34	0.81
CS-VM	74.56	76.39	0.835
MTL-DL	76.52	79.24	0.883
HDL	72.86	76.52	0.92
FLSTM-ALM	98.83	99.32	0.99

Testing

Model	Accuracy	F1-Score	QWK
CA-DL	82.66	84.23	0.82
CS-VM	74.9	77.24	0.856
MTL-DL	75.9	78.51	0.88
HDL	73.59	78.2	093
FLSTM-ALM	98.97	99.51	0.992

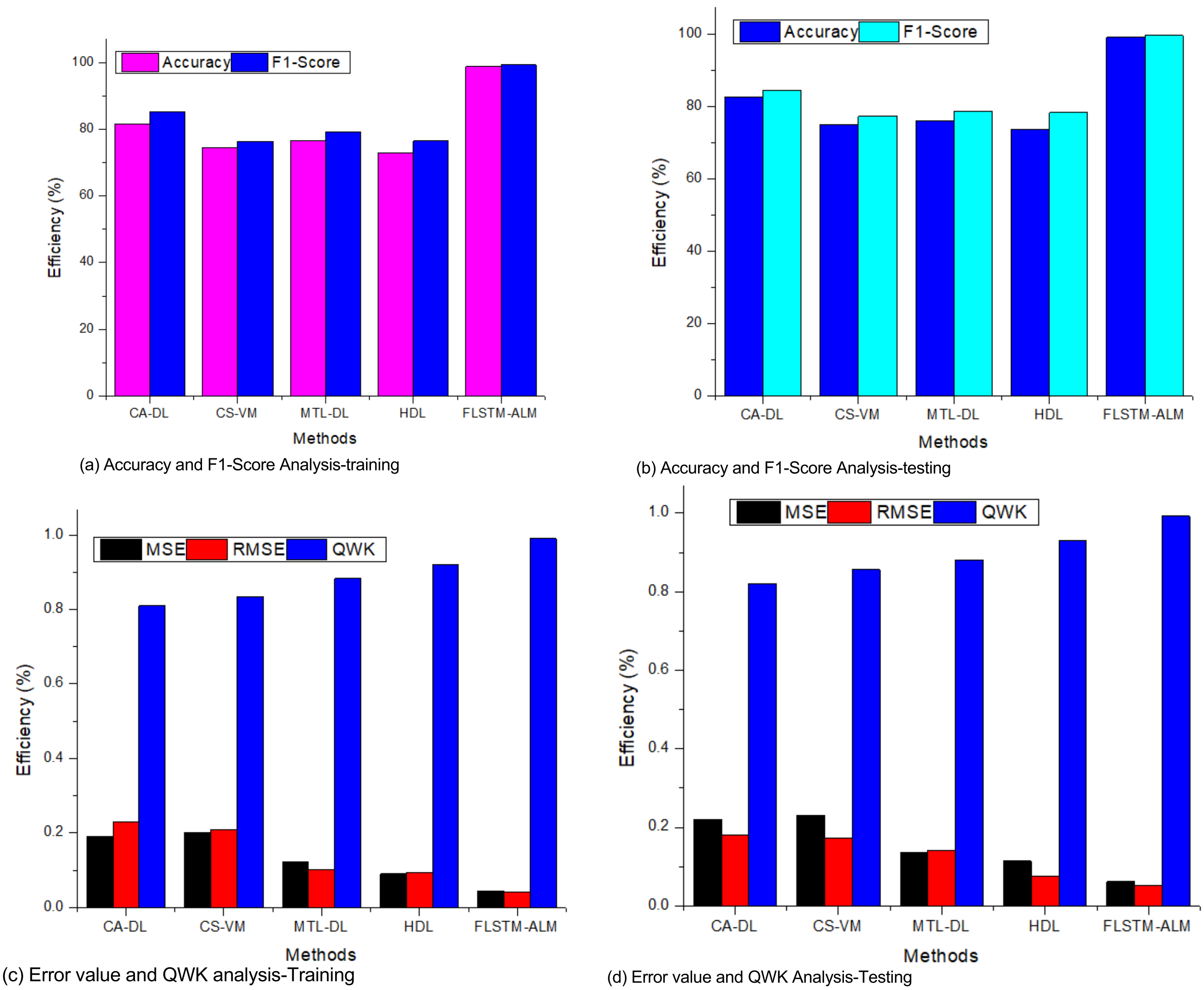
TABLE 3. Performance of error value analysis.

Model	Training		Testing	
	MSE	RMSE	MSE	RMSE
CA-DL	0.19	0.23	0.22	0.18
CS-VM	0.202	0.21	0.23	0.173
MTL-DL	0.123	0.102	0.135	0.142
HDL	0.09	0.095	0.114	0.075
FLSTM-ALM	0.045	0.041	0.062	0.053

Table 2 illustrates the efficiency of the FLSTM-ALM-based automatic essay score prediction process. The results indicate that the introduced approach achieves 98.97% accuracy, directly showing that the method correctly classifies new easy features with a minimum deviation error. The model uses the masked language model, which guesses the word according to a high probability value. Then, the neural knowledge retriever and augmented encoder are applied to predict the retriever's learning pattern. The extracted learning patterns and optimised feature selection process minimise the classification error rate and reduce irrelevant feature involvement. The effective utilisation of the optimisation algorithm improves the overall recognition efficiency by up to 99.51%. In addition, the system achieves a maximum QWK value of 0.992 compared to the existing approach.

As per the guidelines in Table 3, the system produces more perfect documents than the hand-written score document. The system uses the optimisation algorithm while updating or fine-tuning the network parameters. The successful updating process minimises deviation errors such as MSE and RMSE. The minimum error values show that the system ensures a high classification rate. The overall results are shown in Table 3

Table 3 illustrates the error rate analysis of the FLSTM-ALM approach-based essay scoring system on training and testing. It clearly shows that the system ensures the minimum error values (MSE-0.045 and RMSE-0.041), and compared to other methods such as CA-DL (MSE-0.19 and



(a) Accuracy and F1-Score Analysis-training
(b) Accuracy and F1-Score Analysis-testing
(c) Error value and QWK analysis-Training
(d) Error value and QWK Analysis-Testing

FIGURE 5. Efficiency analysis of FLSTM-ALM.

RMSE-0.23), CS-VM (MSE-0.202 and RMSE-0.21), MTL-DL (MSE-0.123 and RMSE-0.102) and HDL (MSE-0.09 and RMSE-0.095). The language model uses the LSTM as the neural model, with three gates for computing the output value. The network has cell states, hidden states, and memory states, which are utilised to save the information while processing the output. In addition, the network uses an optimisation algorithm to update network parameters and minimise the error rate. Figure 5 provides a graphical analysis of the table's discussions.

Performance measures are essential for assessing the efficacy and correctness of machine learning models, particularly those used for automated essay scoring, classification, and regression. AES model performance can be assessed using the MAE measure, frequently employed in regression issues. In contrast to recall, which is used to assess the model's capacity to identify positive and negative examples in the dataset accurately, accuracy assesses the proportion of correct predictions provided by the model. Hence, these metrics can be modified to assess AES models' performance and are frequently utilised in classification and regression issues.

Figure 6 depicts the comparison analysis of the efficiency measure of the proposed model using recall and accuracy. The proposed model outperforms the CS-VM, CA-DL, MTL-DL, and HDL algorithms. The recall measure is calculated using eqn (25) to judge the accuracy of the AES model's predictions and its overall performance.

$$Recall = TP/(TP + FN) \tag{25}$$

where TP represents the True Positive (TP) correctly predicted scores and FN total count of actual positively predicted scores False Negative (FN) in the dataset.

B. DISCUSSION

Figure 5 demonstrates that FLSTM-ALM modelling achieves more effective results than other methods. The system uses an augmented learning model to train the text features effectively. The model employs a neural knowledge retriever and

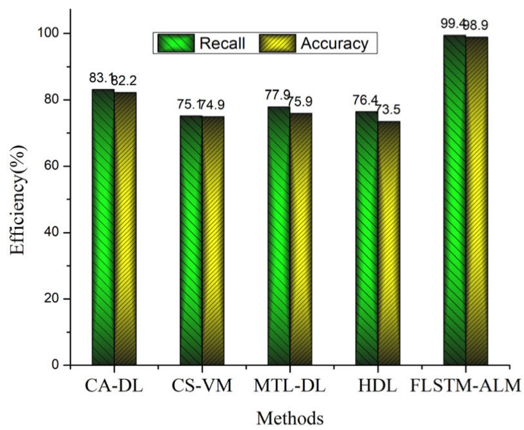


FIGURE 6. **Efficiency analysis of FLSTM-ALM using recall and accuracy measures.**

Output Class	HDL	ABC-BERT-FTM	CS-VM	CA-DL	MTL-DL	FLSTM-ALM
HDL	342 41.0%	2 0.2%	3 0.4%	4 0.5%	1 0.1%	97.2% 2.8%
ABC-BERT-FTM	3 0.4%	211 25.3%	0 0.0%	0 0.0%	0 0.0%	98.6% 1.4%
CS-VM	4 0.5%	1 0.1%	54 6.5%	13 1.6%	3 0.4%	72.0% 28.0%
CA-DL	2 0.2%	1 0.1%	8 1.0%	79 9.5%	0 0.0%	87.8% 12.2%
MTL-DL	0 0.0%	0 0.0%	0 0.0%	0 0.0%	104 12.5%	100% 0%
FLSTM-ALM	97.4% 2.6%	98.1% 1.9%	83.1% 16.9%	82.3% 17.7%	96.3% 3.7%	94.6% 5.4

FIGURE 7. **Confusion matrix analysis.**

encoder to analyse the generated retrieval scores. With these scores, the neural model effectively classifies the output. The Fox optimisation algorithm selects the features based on their relevance to the essay scoring process. This selection of parameters resolves the exploration and optimisation problems while classifying the documents. The system's effectiveness is evaluated using the CM, and the results are shown in Figures 5 (a) and (b).

Figure 7 presents the confusion matrix values for the FLSTM-ALM modelling-based score prediction process. The CM visualization clearly shows that the system effectively classifies input with maximum recognition accuracy, as the BERT model eliminates redundant features before processing. The feature selection procedure creates a template using the selected text, which minimises computation errors. The effective utilisation of the fitness function, local search, and global search processes helps to update the network parameters effectively. The network uses the memory, cell, and hidden state to store processing information, reducing irrelevant information processing and high computation time.

V. CONCLUSION

This research discusses the Fox-optimised LSTM-based augmented language model for predicting the automated score value for essays. First, an augmented model is applied to analyse the inputs, which use the neural knowledge retriever and knowledge augment encoder to identify the learning patterns. The learning patterns are created according to the count-based, morphological, POS tagging, and lemma features. Then, the Fox optimisation algorithm is applied to extract and optimise the features according to their global and local search in the set of features. Finally, the system's performance is evaluated using the experimental results, in which the introduced approach recognises that the essay scores up to98.97 % accuracy.

The accuracy and effectiveness of automated essay scoring systems can be improved using the proposed FLSTM-ALM approach and a neural knowledge encoder and retriever. First, an augmented model analyses the inputs and detects learning patterns using a neural knowledge retriever and knowledge augment encoder. The count-based features, morphological features, POS tagging, and lemma features are used to generate the learning patterns. After that, the Fox optimisation method was used to extract and optimise the features in the set of features based on their global and local searches. The output is then classified using the LSTM neural and enhanced models. Finally, the system's effectiveness is assessed using the experimental findings, demonstrating that the introduced approach can accurately identify up to 98.97% of essay scores.

An advantage of the FLSTM-ALM model is that it employs an augmented learning model, which enhances the standard of the training set of textual characteristics. The model makes it easier to get around problems with conventional feature extraction techniques, such as how expensive it is to process and train the data. The major limitation is considering the computing expenses for training and deploying the proposed model in practical situations. The future scope could assess the technique's effectiveness when implemented for essays written in other languages or on subjects unrelated to those addressed by the ASAP and ETS databases.

REFERENCES

[1] S. M. Eid and N. M. Wanas, "Automated essay scoring linguistic feature: Comparative study," in *Proc. Int. Conf. Adv. Control Circuits Syst. (ACCS) Syst. Int. Conf. New Paradigms Electron. Inf. Technol. (PEIT)*, Nov. 2017, pp. 212–217.

[2] D. A. Abduljabbar and N. Omar, "Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination," *J. Theor. Appl. Inf. Technol.*, vol. 78, no. 3, p. 447, 2015.

[3] M. Mohammed and N. Omar, "Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec," *PLoS ONE*, vol. 15, no. 3, Mar. 2020, Art. no. e0230442.

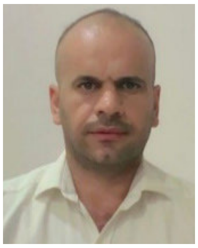
[4] M. T. Alshammari, "Evaluation of gamification in E-learning systems for elementary school students," *TEM J.*, vol. 2, pp. 806–813, May 2020.

[5] C. Li, L. Lin, W. Mao, L. Xiong, and Y. Lin, "An automated essay scoring model based on stacking method," in *Proc. IEEE 2nd Int. Conf. Softw. Eng. Artif. Intell. (SEAI)*, Jun. 2022, pp. 248–252.

[6] J. S. Tan, "Ablation study on feature group importance for automated essay scoring," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 11, no. 1, pp. 90–101, Jun. 2022.

[7] R. H. Chassab, L. Q. Zakaria, and S. Tiun, "Automatic essay scoring: A review on the feature analysis techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 10, pp. 1–14, 2021.

[8] R. A. Rajagede, "Improving automatic essay scoring for Indonesian language using simpler model and richer feature," *Kinetik: Game Technol., Inf. Syst., Comput. Netw., Comput., Electron., Control*, vol. 6, pp. 11–18, Feb. 2021.
[9] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: A systematic literature review," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2495–2527, Mar. 2022.
[10] M. Suhaidi, R. A. Kadir, and S. Tiun, "A review of feature extraction methods on machine learning," *J. Inf. Syst. Technol. Manage.*, vol. 6, no. 22, pp. 51–59, Sep. 2021.
[11] E. Mayfield and A. W. Black, "Should you fine-tune BERT for automated essay scoring?" in *Proc. 15th Workshop Innov. Use NLP Building Educ. Appl.*, 2020, pp. 151–162.
[12] M. A. Almaiah and I. Y. Alyoussef, "Analysis of the effect of course design, course content support, course assessment and instructor characteristics on the actual use of E-learning system," *IEEE Access*, vol. 7, pp. 171907–171922, 2019.
[13] M. Hendre, P. Mukherjee, R. Preet, and M. Godse, "Efficacy of deep neural embeddings based semantic similarity in automatic essay evaluation," *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1379–1389, Dec. 2021.
[14] R. M. Nawi, S. A. M. Noah, and L. Q. Zakaria, "Evaluation of group modelling strategy in model-based collaborative filtering recommendation," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 2, pp. 330–338, Feb. 2020.
[15] H. Faris, I. Aljarah, M. Habib, and P. Castillo, "Hate speech detection using word embedding and deep learning in the Arabic language context," in *Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods*, Valletta, Malta, 2020, pp. 453–460.
[16] M. A. A. Albadr, S. Tiun, M. Ayob, F. T. Al-Dhief, T. N. Abdali, and A. F. Abbas, "Extreme learning machine for automatic language identification utilizing emotion speech data," in *Proc. Int. Conf. Electr., Commun., Comput. Eng. (ICECCE)*, Jun. 2021, pp. 1–6.
[17] H. Zhang and D. Litman, "Co-attention based neural network for source-dependent essay scoring," in *Proc. 13th Workshop Innov. Use NLP Building Educ. Appl.*, 2018, pp. 399–409.
[18] R. Ridley, L. He, X.-Y. Dai, S. Huang, and J. Chen, "Automated cross-prompt scoring of essay traits," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 15, May 2021, pp. 13745–13753.
[19] X. Li, M. Chen, and J. Y. Nie, "SHARED and enhanced deep neural network model for cross-prompt automated essay scoring," *Knowl.-Based Syst.*, vol. 210, Dec. 2020, Art. no. 106491.
[20] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and automated essay scoring," Sep. 2019, *arXiv:1909.09482*.
[21] M. Beseiso and S. Alzahrani, "An empirical analysis of BERT embedding for automated essay scoring," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 204–210, 2020.
[22] L. B. Das, C. V. Raghu, G. Jagadanand, R. A. R. George, P. Yashasawi, N. A. A. Kumaran, and V. K. Patnaik, "FACToGRADE: Automated essay scoring system," in *Proc. IEEE Int. Conf. Ind. 4.0, Artif. Intell., Commun. Technol. (IAICT)*. IEEE, Jul. 2022, pp. 42–48.
[23] R. P. Priya, T. S. Sivarani, and A. G. Saravanan, "Deep long and short term memory based red fox optimization algorithm for diabetic retinopathy detection and classification," *Int. J. Numer. Methods Biomed. Eng.*, vol. 38, no. 3, Mar. 2022, Art. no. e3560.
[24] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13187–13257, Nov. 2023.
[25] H. ALRahhal and R. Jamous, "AFOX: A new adaptive nature-inspired optimization algorithm," *Artif. Intell. Rev.*, vol. 56, no. 12, pp. 15523–15566, Dec. 2023.
[26] M. Khelifi and S. Moussaoui, "ReLM: An efficient reinforcement learning-based localization algorithm for mobile wireless sensor networks," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6.



RIDHA HUSSEIN CHASSAB received the master's degree from the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, where he is currently pursuing the Ph.D. degree. He is a Lecturer with Iraqi Defense University for Military Studies and specializes in natural language processing, computational linguistics, and deep learning.

LAILATUL QADRI ZAKARIA received the Ph.D. degree from the University of Southampton, U.K. She is currently a Senior Lecturer with the Centre of Artificial Intelligence (CAIT), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). She is a member with the Asian Language Processing (ASLAN). Her research interests include natural language processing, computational linguistics, and semantic web technologies.

SABRINA TIUN is currently an Associate Professor with the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). She is a member with the Asian Language Processing (ASLAN) Research Group, Centre of Artificial Intelligence (CAIT). She is the Head of the Postgraduate Program (CAIT), FTSM, UKM. Her research interests include natural language processing, computational linguistics, and speech processing to text analysis.