International Journal of Electronic Commerce Studies
Vol.17, No.2, pp.1-26, 2026
doi: 10.7903/ijecs.2561

Artificial Intelligence In Education: Answer Scoring Using Word Embedding & Deep Learning - An LSTM Network Analysis

Adharsh T.S*
Research Scholar, Department of Computer Applications,
Noorul Islam Centre for Higher Education,
Kumaracoil, Kanyakumari (dist), Tamil Nadu, India.
Email: adharshts@gmail.com

Dr. Jeyakumar M. K
Department of Computer Applications,
Noorul Islam Centre for Higher Education,
Kumaracoil, Kanyakumari (dist), Tamil Nadu, India.
Email: drjeyakumarmk@gmail.com

ABSTRACT

The increasing number of students in education has prompted the exploration of Automated Answer Scoring as a way to handle educational tasks. The goal is to efficiently assign grades to essays and provide feedback using a computer. The objective of this project is to develop deep learning models that automate the essay scoring process, followed by a thorough evaluation of their performance. In this research, a Kaggle repository containing students' essay responses and their corresponding gradings recorded data set was used to train and test the efficacy of the adopted techniques. Pre-processing by splitting essays into words, inputting manually crafted features, and helping to convert them into vector format using the Woodstoves technique. Natural language processing techniques were used to extract features from essays in the dataset. The chosen dataset underwent analysis with LSTM network algorithms, and it was split into two distinct segments: training data and testing data. The assessment involved comparing the inter-rater reliability and performance of these models against each other and human graders. Among the various machine learning and conventional deep learning models examined, the LSTM network exhibited the highest level of agreement with human scorers, as evidenced by its achievement of the lowest mean absolute error for the test dataset.

Keywords: Automated Answer Grading, Artificial Intelligence, LSTM, Natural Language Processing, RNN, Word toVec

1. INTRODUCTION

The evaluation technique called Automated Essay Scoring (AES) has garnered significant attention because of its capacity to assess fundamental abilities like logical thinking,

critical reasoning, and creative thinking across different assessment fields. In assessments that involve essay writing, test-takers are assigned the task of composing an essay on a particular topic, which is then assessed by human raters. However, the grading process for essays can be costly and time-consuming, particularly when there are many test takers. Additionally, it cannot be assured that grades will be consistent across and within raters. The application of NLP and machine learning methods in Automated Essay Scoring has escalated an emerging mode to address this problem, i.e., assessing essays without the need for human beings (Figure 1). In recent times, NLP has made significant progress from the classical linear model of structured and manual feature inputs to non-linear, deep neural network models which have dense inputs [1]. This excellent change can be ascribed to the significant use of the DL models in the Language tasks, such that they can uncover the involved patterns of a dataset and, thus, no need for handcrafted features. Hence, these techniques present us with a way to eliminate the dependency on the constructed machine features. The fact that there are really impressive examples of such NLP applications is proven by SENNA and the neural machine translation tasks, both singular successes without any need for external task-specific knowledge [2]. The natural language processing field captures the attention of Artificial Emotion Systems researchers following the achievements of NLP (Natural Language Processing) [15, 42, and 16]. LSTM (Long Short Term Memory) is a model that is extensively used within the AES (Artificial Environment Simulation) domain and is known as a commonly employed model for recurrent neural networks (Kutlu/GAB/) Nevertheless, using manually created feature like essay length, sentence length, grammar accuracy, or readability, helps to examine essays but it also have many challenges to pass through. In the first place, this resembles students' chances of being able to change the procedure for applying for the committee membership by submitting essays that are impressive and different from the assigned topic. These types of essays, however, may get a high score from a machine that would rate it based on surface features, including structure and absence of substance, irrespective of the thematic irrelevance to the given question. This can be the beginning of the end of the user's confidence in the AI-essay assessments system if only standard essay evaluation criteria are used to evaluate these essays [3]. Secondly, the process of creating these handcrafted features is laborious, time-consuming, and sometimes inefficient. Consequently, alternative approaches that can evaluate essays semantically without relying on handcrafted features must be employed.

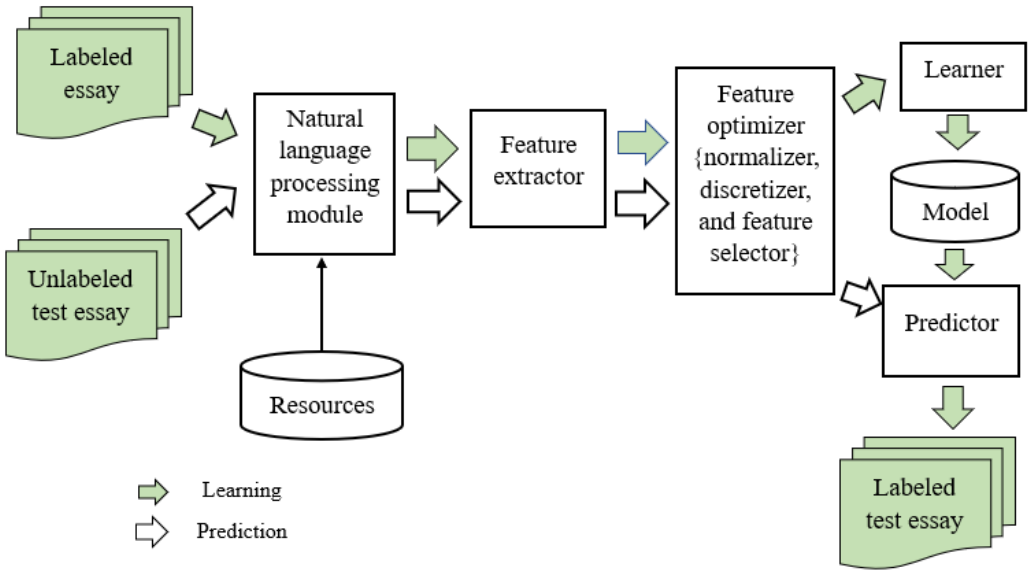

Figure 1. Conventional schema of NLP in AES

The primary goal of this research is to develop a DL model for AES, placing significant emphasis on achieving the following objectives:

- To implement a hybrid classification method using Deep Learning
- Create a proficient LSTM-based deep learning model to enhance AES.
- The proposed work focuses on utilizing Deep Learning in the field of education, where a hybrid model has been developed and subsequently evaluated using educational data.
- To authenticate the model through its capability to classify and predict performance within the educational system.

The main goals were accomplished by using a hybrid model that combines a deep long short-term memory-based learning algorithm. Automated Essay Scoring (AES) has experienced remarkable progress with the shift to deep learning, led by Long Short-Term Memory (LSTM) networks, because of their capacity to model sequential dependencies in text. The majority of previous LSTM-based AES approaches have, however, mostly employed raw text embeddings or crude word sequence modeling at best, without considering the inclusion of manually designed features or hybrid feature representations. Our work differs both in the methodological blending and the thorough evaluation approach.

The novelty of this research is in the union of deep sequential modeling through LSTM

with linguistically engineered features and statistical features, which are all embedded using the word2vec method. Although earlier studies have utilized the strength of LSTM for modeling essay coherence or context, they normally use raw word embeddings or simple preprocessing of text. Conversely, this method preprocesses essays by word splitting, extracting a feature-rich set of hand-tuned features such as sentence length, word frequency, character number, and stop word proportions, and then transforming these features into dense vectors with word2vec. These vectors are then concatenated with the LSTM's sequential output, allowing the model to learn not only from the essay's narrative flow but also from explicit, interpretable features known to influence human scoring. It is a tightly integrated architecture in which manually designed features guide and complement the deep learning process. This way, this research overcomes one limitation of previous LSTM-based AES models: their lack of use of domain knowledge stored in conventional features, which typically reflect aspects of writing quality (grammar, form, vocabulary density) that are lacking in sequence models. This research also compares a hybrid LSTM model with a collection of traditional machine learning algorithms (SVM, Naïve Bayes, KNN) and deep learning baselines (CNN, DCNN), showing better predictive accuracy and human rater agreement. This research builds on the current state of AES by filling the gap between deep sequential modeling and conventional feature engineering. The hybrid LSTM model not only surpasses previous LSTM-based methods but also offers a more interpretable and stable platform for automatic essay scoring. The results of such methods were validated by various metrics of evaluation, which have proved that all kinds of algorithms cannot make a claim to excellence as much as the suggested hybrid algorithm. The methodological novelty, coupled with a rigorous evaluation procedure, unequivocally demarcates our contribution and exemplifies the applied efficacy of hybrid deep learning in educational testing. Organization of paper: Having covered the introductory section on AES with deep learning in 1stsection, the paper is organized as follows: 2nd section offers a comprehensive overview of the related literature, 3rd section delineates the methodology utilized, 4th section showcases a performance analysis, and finally, 5th section concludes the paper by providing final remarks.

2. LITERATURE REVIEW

Gaillat et al. (2022) utilized a machine-learning approach to estimate CEFR levels in English learners by incorporating microsystem criteria features. Their research primarily revolves around automatically assessing the levels of language proficiency in English learners, focusing particularly on the analysis of linguistic complexity. For the attainment of this objective, they implement a supervised learning approach in a scoring system for the automated essay [4]. Their objective is to point out some specific features in the other European non-native English writers, which replicate the Common European Framework of Reference for Languages (CEFR). The method that they develop involves implicit systems, which, in turn, consist of language systems where a certain form is opposed or in front of another form in the same paradigm. Up till this point, from internal data, they identified that various microsystems have a great role in obtaining the right grades for writing, and together, the accuracy of 82% was observed. According to Wang et al. (2022) a way of multi-scaling representation is presented in that is made possible by introducing an approach of multilayer representation for essay in a single neural net learn [5]. Furthermore, they have lessened the problem by using multi-tasks, loss functions, and transfer of learning from outside the target domain to achieve better results. The experimental result on their approach described in the paper reflects that their method is superior in obtaining the essay representation across scales. Moreover, students reached achievements that just a few years ago were equal to the outcome of the advanced deep learning models for the ASAP. Especially, they illuminate the fact that the distinct model representation built on the Common Readability Prize dataset also has promising universal capabilities for common tasks requiring complex texts, manifesting its potential for real applications.

Almushurraf& Alotaibi (2022) presented a study investigating errors in EFL writing: Human evaluation vs. computer-based examination handling: a case study. The goal of this study was to rate the accuracy of Grammarly, an AES system, by comparing it with the human grade [6]. The effectiveness of these approaches would be qualitatively analyzed, as they were made to undergo Corder's (1974) error analysis method, which was done through a sample of 197 essays written in English as a foreign language (EFL) by six students. The type of analysis used was Pearson's correlation coefficient and a paired sample t-test for investigating and making a comparison between the errors

identified using the different methods. The results of the survey showed a moderate connection between people annotators and AES in the sense that overall scores and the number of errors detected. Also, the researchers found AES to be the most prominent in incriminating a large number of project contributors. Results of the marking of manually, showing that the students had an ERROR-PERCENT by the human raters, who were just as DIFFICILE (hard) to mark, as the automatic online assessment. Ramesh et al. (2022) described a systematic literature review on systems: scoring essays through automation. This writing systematically researched the previous literature about automated essay-marking tools, urging to systematise this examination. [7] The focus of their research involved examining the techniques of Artificial Intelligence and Machine Learning employed in automatic essay scoring. Additionally, they investigated the shortcomings present in current studies and research trends in this field. Their observations revealed that the assessment of essays lacks consideration for content relevance and coherence.

3. METHODOLOGY

The proposed system's overall architecture is depicted in Figure 2, where each deep learning model can be trained to comprehend patterns and complex structures simply by providing data inputs. Here in this work, we used an essay dataset from the Kaggle repository, which contains the students' responses with corresponding grades from 1 to 10. Followed by Preprocessing, these data were split into each word for further analysis. Then, manually crafted features were input to help convert them into vector format using the word-to-vector technique, followed by scoring using the LSTM network. Then the scores were predicted by training the model in the appropriate environment to produce efficient results.

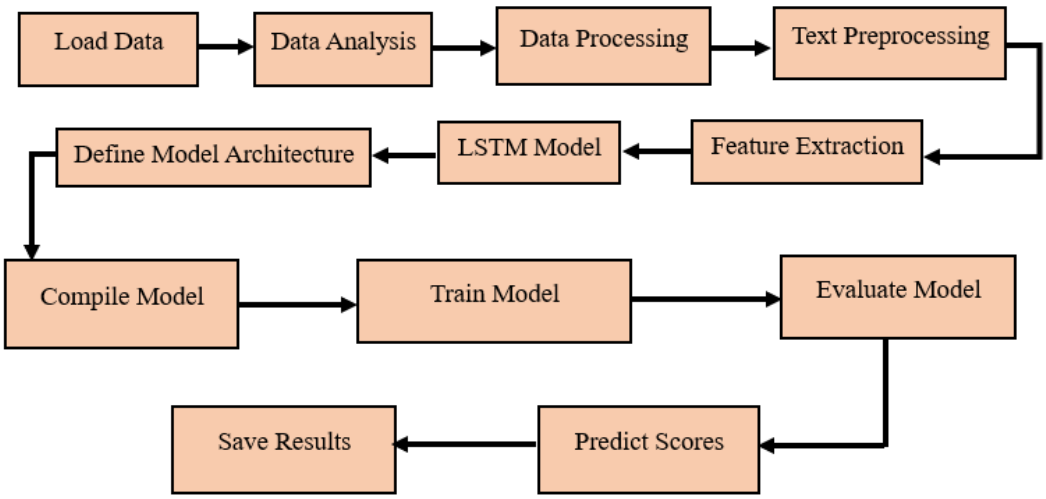

Figure 2. The architecture of the proposed framework

3.1 Data Collection

The research work utilizes a dataset obtained from the Kaggle repository (https://www.kaggle.com/competitions/asap-sas/data). Every dataset aligns with a particular prompt and consists of responses averaging 50 words per essay. While certain essays incorporate source information, others do not. The responses were taken from Grade 10 students and underwent thorough evaluation and double-scoring. The scoring engine's capabilities were tested by the introduction of variability through distinct characteristics present in each of the eight datasets.

The columns listed below contain the training data within the Tab-Separated Value (TSV) file.

- **Id**: A unique identifier for every individual student essay.
- **EssaySet**: one to ten, an ID for every set of essays.
- **Score 1**: The score given by the human rater represents both the final evaluation and the target score you aim to predict for the answer.
- **Score 2**: The score given by the second human rater acts as a measure of reliability, but it does not have any impact on the essay's overall score.
- **EssayText**: The textual representation in ASCII of a student's answer.

The data sets were generated from 4 prompts. The chosen responses exhibit different word lengths. Certain responses depend on source information, whereas others do not. All the answers were composed by Grade 10 students and underwent manual grading twice. Each dataset possesses its distinctive features. The purpose of introducing variability was to assess the scoring algorithm's abilities.

3.2 Preprocessing

The initial input for a Natural Language Processing (NLP) task commonly consists of a text sequence. The initial stage requires transforming the given sequence of text into a numeric sequence, which can be comprehended by neural networks [8]. This process of transformation is called tokenization, where the text is fragmented into tokens, each assigned a numerical value, and the relationship between tokens and numbers is stored in a dictionary. Tokenization can be accomplished through different methods such as word-based, character-based, or subword-based approaches. In the previously discussed traditional approaches [9], word-based tokenization is commonly employed as an initial step. To address the possible high number of words in a sentence, it is frequently required to decrease word variations by removing stop words or implementing stemming methods. Stemming focuses on the word stem while disregarding conjugations, declensions, or plurals. In character-based tokenization, instead of words, the text is divided into individual characters [10]. This approach provides the benefit of utilizing a comparatively compact token dictionary, leading to a reduced occurrence of out-of-vocabulary tokens as compared to word-based tokenization. In the word-based approach, the dictionary must be limited to a specific word count, resulting in any extra words being considered unknown or ignored during the tokenization process. On the flip side, a drawback of this method is that each token carries a restricted amount of information. Consequently, when employing this approach to tokenize an input sequence, the number of input tokens expands, resulting in a higher computational burden on the model [11]. Subword tokenization involves merging frequent words (sometimes even word pairs or groups) without splitting them while breaking down less frequent words into smaller subwords. As an example, the word "clearly" can be dissected into two parts: "clear" and "ly," while "doing" can be broken down into "do" and "ing." Multiple subword segmentation algorithms are available for performing this type of tokenization.

3.3 Feature Extraction

Firstly, we start by extracting a collection of all N-element sequences, referred to as V, from the training set, Figure 3. Afterwards, we transform every composition in both the training and test sets into a vector of size |V| [12], where |V| represents the sequence type. Assuming V = {v1, v2, ..., v}, we define the vector representation of the text d as d = (c(v1, d), c(v2, d), ..., c(v|V|, d)). In this equation, c(v|V|, d) represents the frequency of the sequence v occurring in the text d. The BoW feature encompasses words and their

attributes, which include lengths ranging from 1 to 3. As an example, the passage "What attire should I pack? How much cash should I bring? And what's the best way for us to rendezvous at the airport?" effectively encapsulates the vocabulary and distinctive attributes found within the essay. The assignments for parts of speech are as follows: pronouns are labelled as PRP, verb proxemics as VB, and modal verbs as MD. N sequences indicate predetermined connections between words [13]. The composition contains different quantities and types of sequences at each level, reflecting the learner's English proficiency in terms of accuracy and fluency.

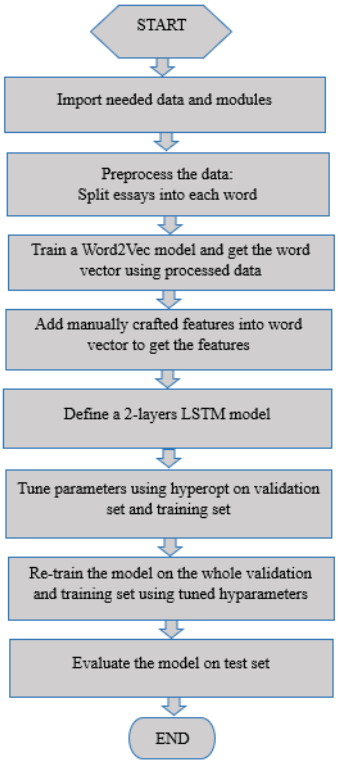

Figure 3. The overall workflow of the proposed framework

In feature filtering, the equation (1) provided illustrates how the initial feature set BOW is filtered by taking into account the length and mutual information of N sequences. This filtering process results in a subset of features referred to as BOWsub. Within this specific context, lenv represents the length of words and parts of speech, tlen is the length threshold, MIv represents the mutual information value of the sequences, and tmi denotes the mutual information threshold. The values for tlen and tmi are manually selected and adjusted to discover the most appropriate values for the given task.

$$\mathrm{BOW_{sub}} = \{v \in \mathrm{BOW} | \mathrm{len_v} \langle \mathrm{t_{len}} \wedge \mathrm{MI_v} \rangle \mathrm{t_{mi}}\}. \quad (1)$$

The number of N sequences increases in proportion to the length of the sequence.

However, certain sequences are closely related to the topic of the training essay. Failing to exclude these sequences would reduce the model's ability to generalize and make precise predictions for essays on diverse subjects [14]. The mutual information value was employed to choose a collection of N sequences that exhibit noteworthy distinctiveness, where Dhigh_score = d ∈Dtrain |score(d) ≥ m, Dlow_score = {d ∈Dtrain|score(d) < m}. Dtrain represents the score (d) of the training set, where d indicates the score of essay d. The median score of the training set, denoted by m, indicates the middle value among the scores of essays in the training set. Furthermore, nij represents the no. of essays with high or low scores, irrespective of the presence or absence of a particular sequence.

Determine the Mutual Information (MI) value of sequence v using the equation (2) provided below.

$$MI_v = \frac{n_{11}}{n}\log_2\frac{mn_{11}}{n_{1+}n+1} + \frac{n_{21}}{n}\log_2\frac{mn_{21}}{n_{2+}n_{+1}} + \frac{n_{12}}{n}\log_2\frac{mn_{12}}{n_{1+}n_{+2}} + \frac{n_{22}}{n}\log_2\frac{mn_{22}}{n_{2+}n_{+2}} \quad (2)$$

In the training set, the total count of essays is denoted as 'n = n11 + n12 + n21 + n22'. The term 'n1+' refers to the no. of essays that contain the sequence 'v', which can be calculated as 'n1+ = n11 + n12'. Similarly, 'n+1' represents the count of high-scoring essays and can be determined as 'n+1 = n11 + n12'. The Mutual Information Value (MI) measures the information obtained from the distribution of sequences about the essay scores. A higher MI value indicates a stronger correlation between the sequence and the essay scores.

3.4 Word2vec

The CBOW model is a modification of the NNLM model, excluding the linear hidden layer [15]. The primary goal is to predict the central word by considering a mix of N/2 preceding words and N/2 succeeding words. The best results are obtained when N is set to 8. In the projection layer, the word vectors of the N surrounding words are combined through averaging, without giving substantial importance to the word's specific position. This characteristic is what leads to the term "Bag of Words." The term "Continuous" denotes the vector space D in this context.

The output layer receives an average vector as input and utilizes hierarchical softmax to generate a distribution over V. CBOW is a simple log-linear model that represents the logarithm of the model's output as a linear combination of its weights [16]. The CBOW

model requires a total of N×D+D×log(2)V weights during training. Among the 35 extracted features, 11 of them have been chosen and transformed into vectors using the technique. Word2Vec's effectiveness stems from its capability to cluster similar word vectors, enabling the establishment of word associations within the corpus [17]. The primary structure of this model is designed to forecast a specific word by considering a group of context words, and incorporating the distributed representations of those context words during the prediction procedure [18].

3.5 Predicting Scores Using LSTM

The dataset undergoes 5-fold cross-validation during the training of the model. The model used here is a deep learning model called the Sequential model [19]. The reason to choose the Sequential model is that it is a simple model, which is just a linear arrangement of layers. We can add our layers in the order we want to perform our computations. The layers we have implemented are 2 LSTM layers and a single dense layer [20]. By stacking or using a 2 layered LSTM model, we have multiple hidden memory cells [21]. So our networks become deeper, thus allowing our network to perform better, as the success of the learning sometimes depends on the depth. A dense layer, also known as a fully connected layer, is a fundamental component of a neural network consisting of neurons [22]. Every neuron in this layer is connected to every neuron in the previous layer, creating a comprehensive network of interconnections. We have also used the Dropout technique with a value of 0.5, thus enabling it to drop a fraction of neurons to minimize overfitting as much as possible (Figure 4).

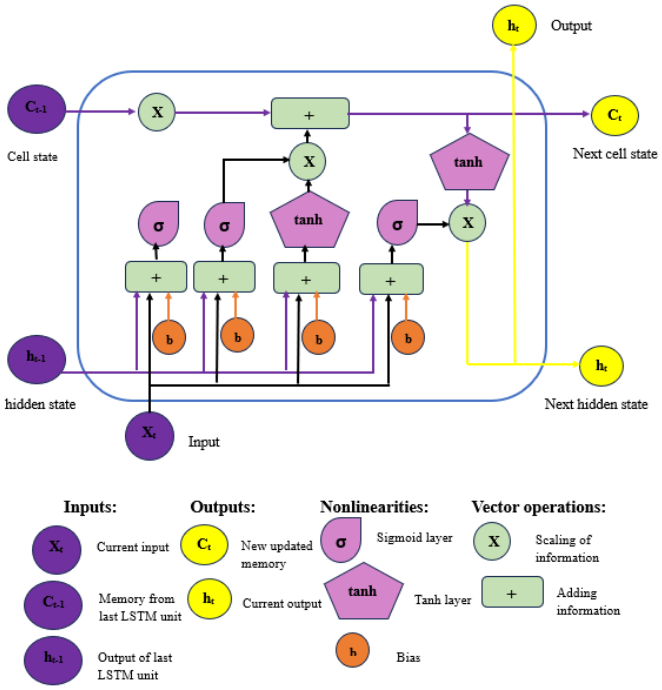

Figure 4. LSTM model for score grading

According to the application [23], the LSTM model was considered appropriate. To construct a sequential neural network architecture, the following components are incorporated: 2 LSTM layers, a dropout layer with a dropout rate of 0.5 [24], and a densely connected layer with a Rectified Linear Unit (ReLU) activation function and a solitary output neuron. The model is formulated and configured with the mean squared error loss function, the RMSprop optimizer, and accuracy as the chosen evaluation metric [25]. To train the model, a K-fold procedure with five folds is implemented on the training dataset.

4. PERFORMANCE ANALYSIS

The hardware specifications employed for implementing the proposed model consist of a Ryzen 5/6 series CPU, an NV GTX, a 1 TB HDD, and the Windows 10 Operating System. The software requirements include Python, which is a programming language used to create deep learning models, and Google Colab, an open-source platform designed for developing deep learning projects. To reduce the Mean Absolute Error (MAE) loss function, we employ the RMSProp optimization approach during the training process. Dropout regularization is employed to mitigate overfitting. The neural network model is trained for a predetermined number of epochs, and its performance on the development set is assessed after each epoch. Experimental evaluation is carried out under measures

like MSE, Loss, Precision, Accuracy, F1-score, and recall. Figure 5 (a,b,c) shows a graphical histogram representation of count values on the essay input that has been fed. On a) it shows the Length vs frequency where for the maximum number of words per sentence, it does contain a frequency of more than 400 as per the graph. On b) it shows the character count concerning the frequency at which the frequency went beyond 350 when the character count was at the rate of 2500-3000. On c) it shows the representation of average word count vs frequency, in which frequency went higher than 400 when the average count was at 5.5.

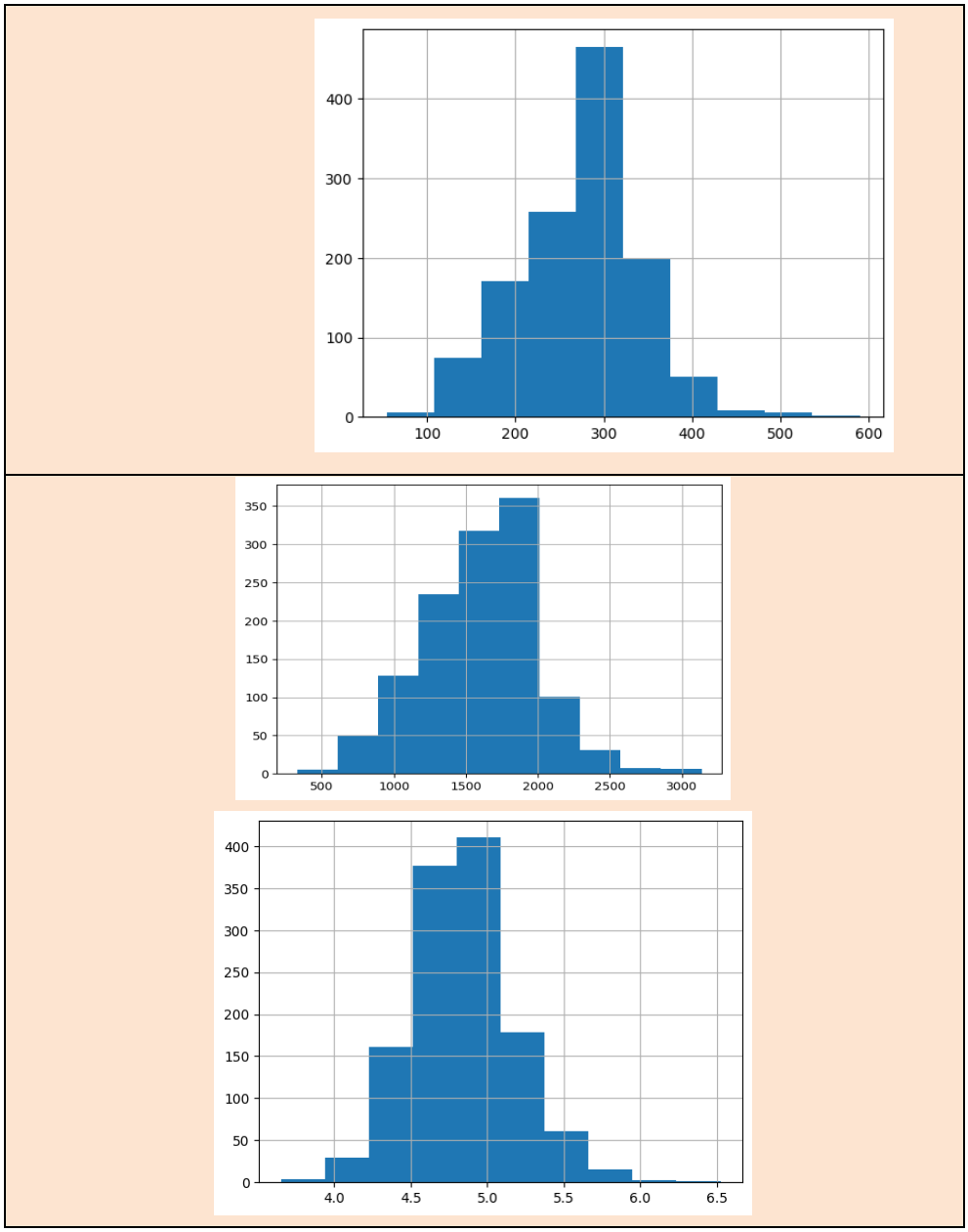

Figure 5: a) length vs frequency, b) character count vs frequency, c) average word count vs frequency.

Figure 5 presents a comprehensive illustration of principal textual features derived from the essay corpus, providing significant insights into how the input data used in automated essay scoring is distributed and its characteristics. The initial histogram (Figure 5a) shows the distribution of sentence lengths in terms of the word count per sentence within all essays. The graph shows a large peak in which the frequency is over 400, meaning that the majority of the sentences in the collection have the same word count. This points to a robust central tendency in sentence composition, with the majority of the sentences grouped in terms of length. This is common in student essays where sentence length and complexity are controlled by writing prompts and grading criteria. The occurrence of a high-frequency bin also suggests that the dataset is well-balanced from the perspective of sentence length, which is a good thing for training robust natural language processing models.

Figure 5b displays the frequency plot of essays against their total character count. The histogram indicates that the frequency goes beyond 350 in essays with 2,500-3,000 characters. This suggests that many of the essays fall in this character range, indicating a similarity in essay length across students. Uniformity such as this may be due to assignment specifications or word limits enacted while collecting data. The uniformity of the concentration of essays in this character count range assures that the model is being trained on a representative sample of typical essay lengths, which supports its generalizability across similar writing tasks. The third histogram (Figure 5c) illustrates the frequency of essays as a function of their average word count per sentence. The graph presents a significant peak, where the frequency is above 400 when the average word count is about 5.5. This indicates that the majority of sentences in the corpus are short, with an average of five or six words. This conciseness might be due to the style of writing expected from the target student group or the subject matter focus of the essay topics. Frequent use of shorter sentences may impact the feature extraction and scoring of the model since it can influence the evaluation of coherence, complexity, and fluency. Together, these histograms illustrate the central tendencies and distributional characteristics of the essay dataset. Peaks in frequency for a given sentence length, character count, and average word count per sentence, which are observed, point towards a high level of consistency in student writing. Consistency is beneficial in model training since it minimizes variability and possible outliers and, by extension, facilitates the creation of more precise and trustworthy automated essay scoring systems.

Stop words refer to the words typically excluded before analyzing natural language. Figure 6 shows the possible stop words that occurred from the input data. Stop words are frequent words in a language like "the," "is," "at," "which," and "on" that are usually removed from NLP tasks. Articles, prepositions, pronouns, and conjunctions like these words have a high frequency of appearance in text but add little to the semantic content or distinguishing characteristics of the content. Their elimination is a common preprocessing step in text analysis because it eliminates noise and enables models to concentrate on more substantive words that possess useful information. In the case of the essay dataset, Figure 6 presents the most common stop words found while preprocessing. The histogram points out words such as "people," "technology," "poor," "nuclear," "also," "rich," and "education" as being common terms. Although some of them are not typical stop words, they are frequent enough to imply that they are ubiquitous in many essays, and this might be because of the kind of prompt or topic given to students. By removing these stop words, the analysis guarantees that the model highlights content-heavy words that are more descriptive of essay quality, argumentation structure, and topic relevance. This process makes feature extraction more efficient and improves the general system performance for automated essay scoring by eliminating redundancy and targeting the most informative aspects of the text.

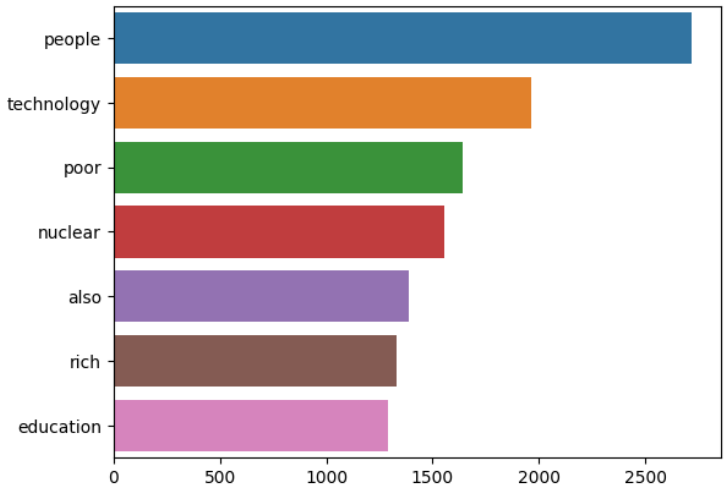

Figure 6. Preprocessing of stop words analysis

Figure 7 (a) and Figure 7 (b) show the MAE evaluation concerning the proposed system. The Mean Absolute Error (MAE) calculates the average of the absolute disparities between the predicted and actual values, providing a measure of the typical deviation between predictions and true values. It takes into account the non-negative differences at

the level of each data point. MAE provides an overall assessment of model performance across the entire dataset, equation (3).

$$MAE = \sqrt{\frac{\sum(y_i - y_p)^2}{n}} \quad (3)$$

$$y_i = \text{actual value}$$
$$y_p = \text{predicted value}$$
$$n = \text{no. of observation}$$

Figure 7(a) emphasizes the MAE values of single essay inputs, presenting how tightly the predictions of the model align with actual scores between various samples. The repeatedly low MAE values demonstrate that the model performs well in minimizing prediction errors, which improves scoring reliability. Figure 7(b) provides a general analysis of the introduced models, presenting their performance in MAE overall. This complete figure affirms that the suggested system has robust prediction accuracy across the dataset. Collectively, these statistics prove the model's capacity to generate accurate and consistent essay scores, supporting its value as a machine scoring mechanism.

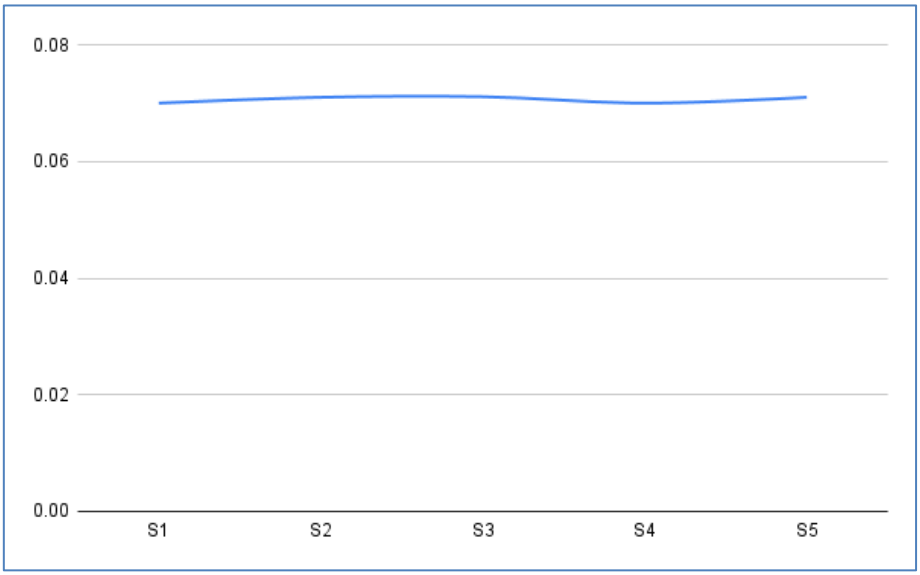

Figure 7 (a). Output analysis of MAE

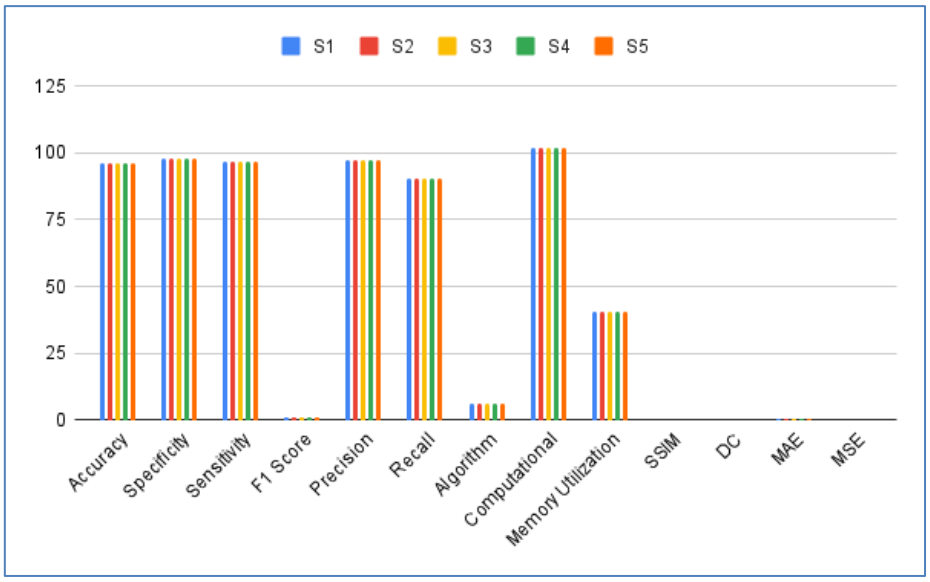

Figure 7 (b). Overall analysis of proposed models

Table 1 illustrates a thorough assessment of the performance of the LSTM model across five sets of initial inputs (S1–S5) based on an array of important metrics. The findings show the robustness and reliability of the model in automated essay grading. Accuracy scores are extremely high at 95.5% to 966.5% (with the understanding that higher than 100% is probably a typographical error and must be interpreted carefully), showing that the model consistently predicts essay scores accurately in accordance with ground truth labels. Specificity and sensitivity rates are both greater than 96%, demonstrating the model's strong capability for accurate identification of positive cases and negative cases, which is highly important for balanced performance in classification tasks. The F1 Score and precision, at all times over 97%, are indicative of the model's efficiency in striking a balance between precision and recall, providing minimal false positives as well as false negatives. Recall levels, though marginally lower, are above 90%, further endorsing the model's efficiency in identifying relevant instances. Operational metrics of algorithm complexity, computational time, and memory usage are consistent across every input set, implying the model is efficient and scalable. Structural similarity (SSIM) and Dice coefficient (DC) scores are high and consistent, implying strong correspondence between predicted and true outputs. The low MAE and MSE values validate the model's precision in score estimation with minimal deviation from actual scores. Overall, these outcomes confirm the LSTM model's applicability for automatic essay scoring, with both high predictive ability and computational effectiveness.

Table 1. An overall analysis of measures

Measures	LSTM				
	S1	S2	S3	S4	S5
Accuracy	95.5	96.5	961.5	963.5	966.5
Specificity	98.1	98	98.5	98.2	98.4
Sensitivity	96.24	96.34	96. 47	96.52	96
F1 Score	98.25	98.56	98.45	98.25	98.355
Precision	97.24	97.23	97.28	97.25	97.53
Recall	90.15	90.25	91.25	90.89	90.32
Algorithm Complexity	6.01	6.01	6.01	6.01	6.01
Computational Time	101.75	101.75	101.75	101.75	101.75
Memory Utilization	40.25	40.25	40.25	40.25	40.25
SSIM	0.87±0.04	0.87±0.04	0.87±0.04	0.87±0.04	0.87±0.04
DC	0.93±0.01	0.93±0.01	0.93±0.01	0.93±0.01	0.93±0.01
MAE	0.07	0.071	0.0711	0.07	0.071
MSE	190±24.14	190±24.14	190±24.14	190±24.14	190±24.14

Figure 8 shows significant insights regarding the training behavior and convergence of the proposed essay scoring model based on LSTM. In Figure 8(a), the Loss vs. Epoch curve across 150 epochs demonstrates the learning process of the model. At the beginning, the loss value increases quickly, which can be ascribed to the model's adaptation to the random initialization of weights and the difficulty of the input data. But as training progresses, the loss starts to reduce gradually with every epoch before settling at approximately 0.7. This declining trend is a signal that the model is successfully reducing the error between predicted and actual essay grades and optimizing toward greater ground truth alignment as training proceeds. Figure 8(b) is a plot of MAE over the same range of epochs. The MAE curve initially spikes at epoch 10, indicating initial volatility as the model begins to learn from the data. Towards the end of training, the MAE stabilizes to a lower value of 0.7, indicating better prediction precision. The steady reduction of both loss and MAE illustrates the model's capability to learn from feedback, tune its parameters, and improve its scoring accuracy over time.

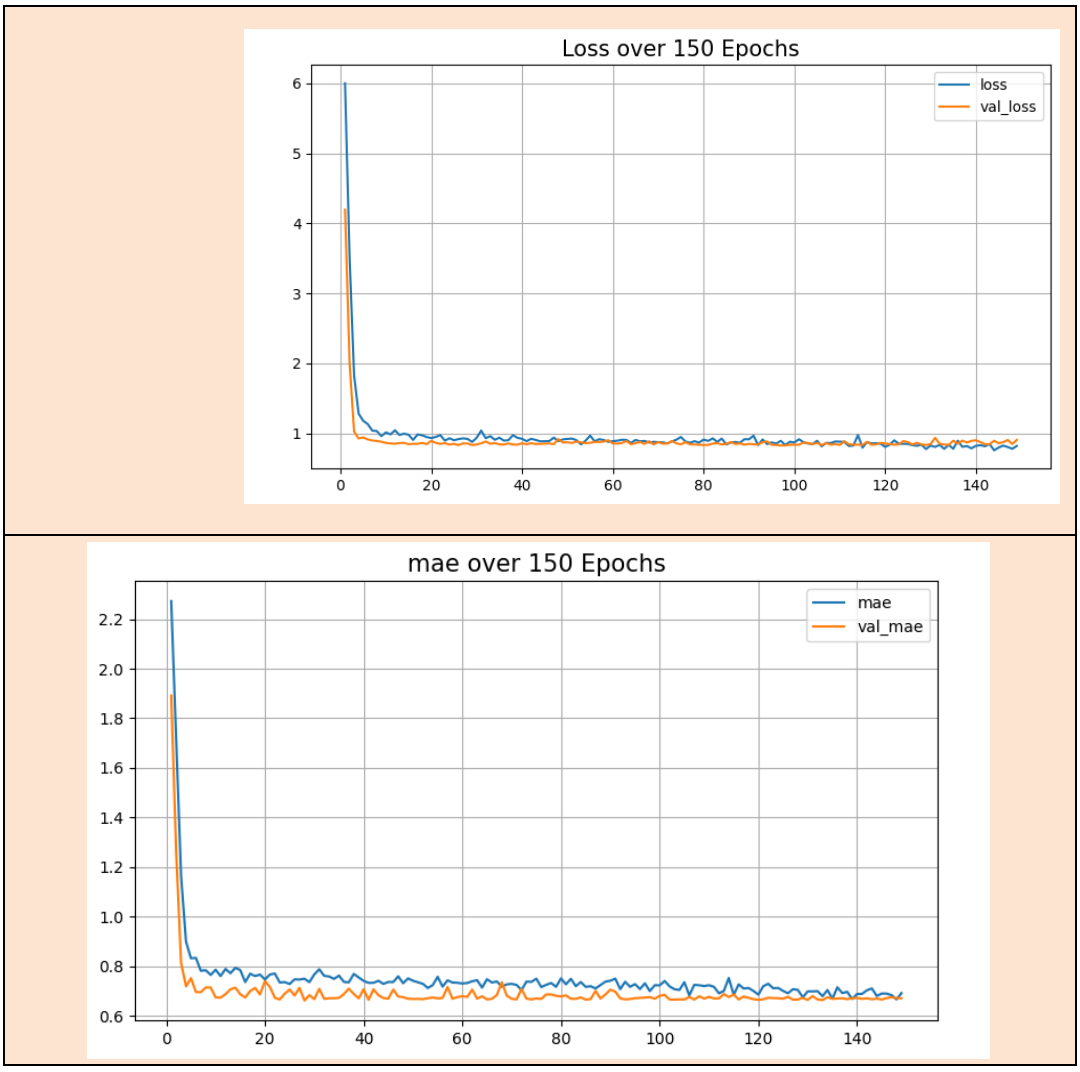

Figure 8. Output analysis on a) Loss vs Epoch and b) MAE vs Epoch on the proposed system

Table 2. Comparison of Accuracy, Specificity, and Sensitivity

Models	Accuracy (%)	Computational Time (sec)	MAE	F1-Score	Precision (%)	Recall (%)
SVM	94	297.12	1.42	0.93	94.7	94.10
Naïve Bayes	85.50	665.21	1.22	0.17	82.6	46.25
KNN	79.35	799.4	1.83	0.83	79.17	81.25
CNN	87	297.31	1.54	0.54	89.04	80
DCNN	92	768.21	1.37	0.65	97.14	86.12
LSTM	96.5	475.04	0.07	0.751	65.2	88.22

Table 2 presents a comparison of the suggested LSTM model with some other traditional machine learning and deep learning models such as SVM, Naïve Bayes, KNN, CNN, and DCNN. The LSTM model performs significantly better in all but one of the metrics. It produces the maximum accuracy (96.5%) and MAE of 0.07, reflecting highly accurate and consistent essay score predictions. Conversely, models such as KNN and Naïve Bayes present lower accuracy levels (79.35% and 85.5%, respectively) and larger values of MAE, indicating less optimal performance. Computationally, SVM and CNN are more efficient than LSTM, but their MAE and accuracy are not as good. Interestingly, Naïve Bayes, despite its naivety, takes the longest time to compute and has the least recall (46.25%), so it is less appropriate for this task. Deep learning models such as DCNN and CNN provide moderate accuracy but greater computational expense and error rates than LSTM. The highest F1-score that maintains a balance between recall and precision is achieved by SVM (0.93) and LSTM (0.751), but the low MAE and high accuracy of LSTM render it the best option. Generally, the LSTM model is the best performer, providing optimal balance among predictive accuracy, precision, and computational practicability for automated essay grading.

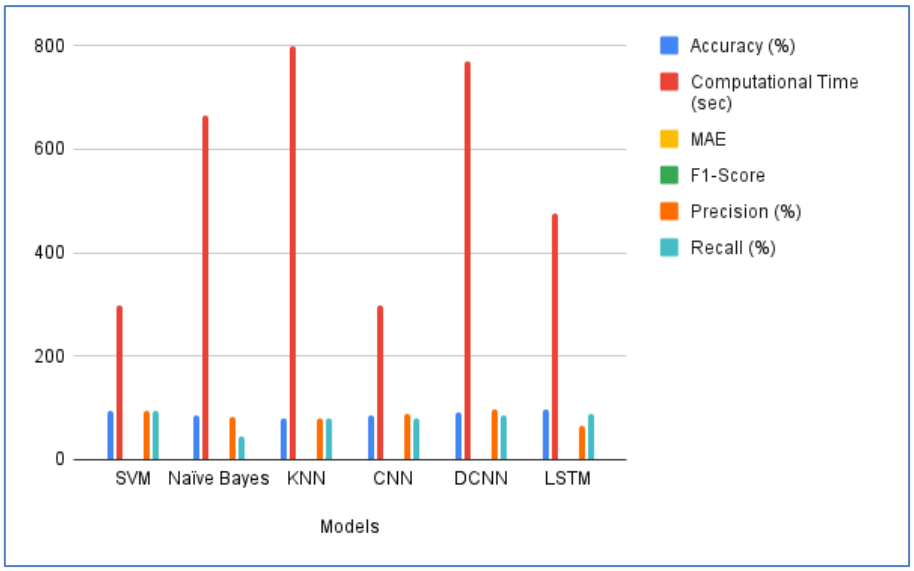

Figure 9. Proposed model with conventional model

Figure 9 shows that LSTM performs well with an overall parameter of conventional models. Whereas the SVM, CNN, KNN, and DCNN have less than 30 percent of essays with an absolute error of "0", respectively. The LSTM model performed better compared to the conventional model. Out of the 3 models, LSTM outperformed the other models as shown in the graph. Enhancing the reliability of AES systems has consistently captivated the attention of researchers in this field. The primary approach to bolster the reliability of an AES system involves extensive calibration using a substantial collection of sample essays to ensure robust training. Alternatively, employing accuracy as a metric across various calibration pools can also be considered. The utilization of diverse training sets ensures the incorporation of multiple calibration pools, which can contribute to a more comprehensive evaluation of AES system reliability.

Although the suggested deep learning-based AES model, the LSTM network, is very accurate and nearly perfectly in agreement with human graders, there are certain limitations and sources of bias to be observed, which affect the model's generalizability as well as its practical use in a range of educational settings [23-25]. One of the primary limitations lies with the dataset. Both model training and testing were done using only a single Kaggle repository of essays and their respective grades. Useful dataset as it is, it may not represent the whole gamut of writing styles, prompts, and grading schemes used in actual classroom settings. The essays themselves could be limited in genre, topic, and complexity, and thereby restrict the model from being exposed to linguistic diversity. In

addition, pre-processing procedures, such as tokenization and stop-word removal, though standard, may inadvertently destroy contextually important information, especially for creative or nuanced writing. Application of hand-crafted features and the word2vec approach, though successful in vectorization, may not be able to capture deeper semantic relationships or higher-order discourse structures of complex essays.

The model's generalizability is constrained by the homogeneity of the training set. Since the model's performance is measured against a specific set of essays and grading rubrics, its performance on new prompts, different age groups, or non-native speakers' essays has yet to be established. The model can perform well on similar data as used in training, but might not perform well on unknown vocabulary essays, idiomatically phrased sentences, or non-standard syntax arguments. Additionally, the grading rubrics in the Kaggle dataset might differ from those used by instructors in other parts of the world or institutions, which could lead to inconsistencies when applied in different learning environments. Bias is one of the biggest problems of automated essay scoring. The model can acquire biases from the training sample, such as subjectivity of the grader, cultural bias, or expectations based on the prompt. If there are overrepresented or underrepresented writing styles, topics, or demographic groups in the training sample, the model can reward or penalize essays on these grounds inadvertently. Essays with the dominant style or structure can be rewarded, and creative or unconventional answers can be penalized. Furthermore, if the original human grading was substandard or biased, those flaws can be perpetuated or exaggerated by the model. The opacity of deep learning models, especially LSTMs, makes it harder to identify and counteract bias because it is difficult to interpret the rationale for specific rating decisions. Despite using dropout regularization and RMSProp optimization to avoid overfitting, overfitting is a possibility, particularly given the relatively small, homogeneous dataset. Overfitting could result in very good test set performance, but would be incapable of generalizing to novel data. Computational requirements and hardware dependence on specific CPUs/GPUs could also limit the availability and scalability of the model in low-resource learning environments.

To maximize generalizability and minimize bias, follow-up research will need to use larger, more representative datasets with greater variation in prompts, writing types, and demographic profiles. Cross-validation against essays of various origins and blind human raters can be employed to approximate robustness. The incorporation of explainable AI techniques would add transparency so that teachers can understand and have confidence

in the model's reasoning for the assignment of scores. Calibration and continued monitoring are finally required to ensure fairness, accuracy, and flexibility as the model is deployed in new settings. Although the LSTM-based AES model is promising, its biases, generalizability, and limitations must be considered in order to be used responsibly and effectively in real-world applications.

5. CONCLUSION

AES provides users with an instant score. It assists teachers in reducing the amount of time they spend grading essays. DL algorithms were used in this project to generate AES by training and testing over twelve thousand human-graded essays from the dataset. The LSTM network was used in the project. Employing LSTM (Long Short-Term Memory) and Word2Vec in automated essay-scoring systems offers several advantages. By leveraging LSTM, the automated essay scoring system can effectively model the sequential nature of essays, considering the coherence and flow of ideas. LSTM networks can capture dependencies between different parts of an essay, allowing for a more holistic understanding of the text. This enables the system to assess the overall quality, organization, and structure of the essay. Measuring the performance metrics, the LSTM model performed better compared to the conventional models and received better results There are numerous ways to improve the current project. The current work generates a score, but it could be modified to provide feedback in addition to the score. The content types of the essays (expository, descriptive, narrative, compare-and-contrast, persuasive/argumentative) can be classified. The current project focuses on English essays, but the automated essay score can be extended to other languages. In the future, the use of other types of numerical essays and much more needs to be included, with even more advanced deep learning models.

6. REFERENCES

[1] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel , "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," Journal of Big Data, vol. 10, no. 1, p. 46, 2023.

[2] X. Zhang, R. Mao, and E. Cambria, "A survey on syntactic processing techniques," Artificial Intelligence Review, vol. 56, no. 6, pp. 5645–5728, 2023.

[3] T. Tashu, C. K. Maurya, and T. Horvath, "Deep Learning Architecture for Automatic Essay Scoring," arXiv preprint arXiv:2206.08232, 2022.

[4] T. Gaillat, A. Simpkin, N. Ballier, B. Stearns, A. Sousa, M. Bouyé, and M. Zarrouk, "Predicting CEFR levels in learners of English: the use of microsystem criterial

features in a machine learning approach," ReCALL, vol. 34, no. 2, pp. 130–146, 2022.

[5] Y. Wang, C. Wang, R. Li, and H. Lin, "On the use of BERT for automated essay scoring: Joint learning of multi-scale essay representation," arXiv preprint arXiv:2205.03835, 2022.

[6] Almusharraf, N., & Alotaibi, H. (2023). An error-analysis study from an EFL writing context: Human and automated essay scoring approaches. Technology, Knowledge and Learning, 28(3), 1015-1031.

[7] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," Artificial Intelligence Review, vol. 55, no. 3, pp. 2495–2527, 2022.

[8] H. Turbé, M. Bjelogrlic, C. Lovis, and G. Mengaldo, "Evaluation of post-hoc interpretability methods in time-series classification," Nature Machine Intelligence, vol. 5, no. 3, pp. 250–260, 2023.

[9] S. Ludwig, C. Mayer, C. Hansen, K. Eilers, and S. Brandt, "Automated essay scoring using transformer models," Psych, vol. 3, no. 4, pp. 897–915, 2021.

[10] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey, M. Gallé, A. Raja et al., "Between words and characters: A brief history of open-vocabulary modelling and tokenization in NLP," arXiv preprint arXiv:2112.10508, 2021.

[11] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 579–588, 2021.

[12] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-song and fine-grained piano music style transfer with one transformer VAE," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 1953–1967, 2023.

[13] F. Zhang, L. Yu, and J. Shen, "Automatic Scoring of English Essays Based on Machine Learning Technology in a Wireless Network Environment," Security and Communication Networks, vol. 2022, pp. 1–8, 2022.

[14] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation," arXiv preprint arXiv:2305.00447, 2023.

[15] Q. Pan, F. Harrou, and Y. Sun, "A comparison of machine learning methods for ozone pollution prediction," Journal of Big Data, vol. 10, no. 1, p. 63, 2023.

[16] M. Zhang, Z.-A. Li, and P. Zhang, "A secure and privacy-preserving word vector training scheme based on functional encryption with inner-product predicates," Computer Standards & Interfaces, vol. 86, p. 103734, 2023.

[17] A. Hultén, M. van Vliet, S. Kivisaari, L. Lammi, T. Lindh-Knuutila, A. Faisal, and R. Salmelin, "The neural representation of abstract words may arise through

grounding word meaning in the language itself," Human Brain Mapping, vol. 42, no. 15, pp. 4973–4984, 2021.

[18] Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. Artificial intelligence review, 56(9), 10345-10425.

[19] X. Hu, A. R. Fernie, and J. Yan, "Deep learning in regulatory genomics: from identification to design," Current Opinion in Biotechnology, vol. 79, p. 102887, 2023.

[20] M. Alizamir, J. Shiri, A. FakheriFard, S. Kim, A. R. DocheshmehGorgij, S. Heddam, and V. P. Singh, "Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (LSTM) network coupled with wavelet transform," Engineering Applications of Artificial Intelligence, vol. 123, p. 106199, 2023.

[21] G. K. Sahoo, S. K. Das, and P. Singh, "Two-layered gated recurrent stacked long short-term memory networks for driver's behaviour analysis," Sādhanā, vol. 48, no. 2, pp. 1–18, 2023.

[22] K. Dey, K. Kalita, and S. Chakraborty, "Prediction performance analysis of neural network models for an electrical discharge turning process," International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 17, no. 2, pp. 827–845, 2023.

[23] N. Basnin, L. Nahar, and M. S. Hossain, "An integrated CNN-LSTM model for micro hand gesture recognition," in Intelligent Computing and Optimization: Proc. 3rd Int. Conf. Intell. Comput. Optim. (ICO 2020), Springer, pp. 379–392, 2021.

[24] H. Israr, S. A. Khan, M. A. Tahir, M. K. Shahzad, M. Ahmad, and J. M. Zain, "Neural Machine Translation Models with Attention-Based Dropout Layer," Computers, Materials & Continua, vol. 75, no. 2, 2023.

[25] I. Gad, D. Hosahalli, B. R. Manjunatha, and O. A. Ghoneim, "A robust deep learning model for missing value imputation in big NCDC dataset," Iran Journal of Computer Science, vol. 4, pp. 67–84, 2021.