


Article

Hybrid Approach to Automated Essay Scoring: Integrating Deep Learning Embeddings with Handcrafted Linguistic Features for Improved Accuracy

Muhammad Faseeh ¹, Abdul Jaleel ², Naeem Iqbal ³, Anwar Ghani ^{4,5}, Akmalbek Abdusalomov ^{6,7},
Asif Mehmood ^{8,*} and Young-Im Cho ^{9,*}

- ¹ Department of Electronic Engineering, Jeju National University, Jeju-si 63243, Republic of Korea
- ² Department of Information Technology, Asia Pacific International College, Parramatta, Sydney 2150, Australia
- ³ Centre for Secure Information Technologies (CSIT), Momentum One Zero (M1.0), School of Electronics Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT3 9DT, UK
- ⁴ Department of Computer Science, International Islamic University, Islamabad 44000, Pakistan
- ⁵ Big Data Research Center, Department of Computer Engineering, Jeju National University, Jeju-si 63243, Republic of Korea
- ⁶ Department of Artificial Intelligence, Tashkent State University of Economics, Tashkent 100066, Uzbekistan
- ⁷ Department of Computer Systems, Tashkent University of Information Technologies Named after Muhammad Al-Khwarizmi, Tashkent 100200, Uzbekistan
- ⁸ Department of Biomedical Engineering, College of IT Convergence, Gachon University, Sujeong-gu, Seongnam-si 13120, Republic of Korea
- ⁹ Department of Computer Engineering, Gachon University, Sujeong-gu, Seongnam-si 13120, Republic of Korea
- * Correspondence: asif@gachon.ac.kr (A.M.); yicho@gachon.ac.kr (Y.-I.C.)

Abstract: Automated Essay Scoring (AES) systems face persistent challenges in delivering accuracy and efficiency in evaluations. This study introduces an approach that combines embeddings generated using RoBERTa with handcrafted linguistic features, leveraging Lightweight XGBoost (LwXGBoost) for enhanced scoring precision. The embeddings capture the contextual and semantic aspects of essay content, while handcrafted features incorporate domain-specific attributes such as grammar errors, readability, and sentence length. This hybrid feature set allows LwXGBoost to handle high-dimensional data and model intricate feature interactions effectively. Our experiments on a diverse AES dataset, consisting of essays from students across various educational levels, yielded a QWK score of 0.941. This result demonstrates the superior scoring accuracy and the model's robustness against noisy and sparse data. The research underscores the potential for integrating embeddings with traditional handcrafted features to improve automated assessment systems.

Keywords: automated essay scoring; RoBERTa embeddings; handcrafted features; LwXGBoost; RoBERTa

MSC: 68T07; 62H30



Citation: Faseeh, M.; Jaleel, A.; Iqbal, N.; Ghani, A.; Abdusalomov, A.; Mehmood, A.; Cho, Y.-I. Hybrid Approach to Automated Essay Scoring: Integrating Deep Learning Embeddings with Handcrafted Linguistic Features for Improved Accuracy. *Mathematics* **2024**, *12*, 3416. <https://doi.org/10.3390/math12213416>

Academic Editors: Shuo Yu, Feng Xia and Jonathan Blackledge

Received: 30 August 2024

Revised: 25 October 2024

Accepted: 29 October 2024

Published: 31 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Grading written essays remains a constant challenge in education, characterized by subjectivity, resource constraints, and the need for expert input. Effective essay grading requires topic knowledge, comprehension of linguistic nuances, organizational coherence, and convincing arguments. Despite ongoing efforts by schools and organizations to standardize and enhance grading systems, the introduction of AI technology marks a new phase of innovation in this domain, potentially redefining AES [1].

AES has sparked interest across academic disciplines, including professionals in education [2,3], NLP [4,5], and linguistics [6]. Traditional techniques in AES rely on manual feature extraction, as opposed to neural network systems that use automatic feature extraction from raw text input. While conventional approaches have proven successful, they

frequently lack flexibility due to their reliance on handcrafted elements. In contrast, neural network algorithms promise flexibility and durability while grading essays. However, DNN success depends on access to vast amounts of labeled data, which is not always readily available.

Current essay scoring systems analyze the quality of written essays mainly through ML models and NLP approaches [7–9]. These methods examine several text characteristics, such as linguistic qualities, coherence, organization, and reasoning. Annotated essay datasets train ML algorithms to uncover patterns and connections between distinct writing features and their associated scores. On the other hand, NLP approaches help to understand the text’s semantic meaning and syntactic structure, allowing for the detection of grammatical errors, vocabulary richness, and other linguistic nuances. While these systems have demonstrated potential for automating essay grading and delivering consistent marks, they also have limitations [10]. These issues include dealing with subjective aspects of writing, such as creativity and originality, and adapting to new writing styles and topics. Furthermore, ensuring fairness and avoiding biases in grading is an important consideration. As technology advances and new approaches emerge, there is a growing demand for more sophisticated essay-scoring methods that can successfully address these challenges while offering accurate and reliable evaluations of student writing.

Exploring pretrained models is a tempting alternative for research, owing to their applicability across multiple applications, which eliminates the need to construct models from scratch [11]. BERT-based models significantly advance pretrained language models, enabling fine-tuning for various tasks while minimizing extensive data labeling requirements. While transformer models have proven successful in multiple fields, their use in AES represents a fresh frontier for innovation.

The proposed method combines handmade characteristics with RoBERTa embeddings within an ensemble framework that notably leverages the LwXGBoost model to predict overall scores. This system, which combines complex RoBERTa embeddings and traditional handcrafted characteristics, uses both methodologies to increase essay grading accuracy and speed due to its lightweight. As a result, it covers current gaps and has tremendous promise for expanding the field of AES.

The paper’s contributions include:

- Development of a Hybrid Feature Extraction Approach by incorporating contextualized semantic information encoded by RoBERTa embeddings and linguistic insights included in handmade features.
- Optimized essay scoring accuracy by fusing features and choosing LwXGBoost for the best QWK score.
- Performed a comparative analysis of BERT, RoBERTa, SVM, AdaBoost, and LwXGBoost models regarding QWK score, MSE, and RMSE for better evaluation.

The structure of the paper is outlined as follows: Section 2 provides an extensive review of the existing literature concerning essay scoring methodologies. Section 3 thoroughly explains the proposed method, presenting the complete framework developed. Section 4 outlines the experimental setup and evaluates the proposed model. Finally, the conclusions drawn from the proposed research are presented in Section 7 of the paper.

2. Literature Review

The AES methodologies were systematically divided into two significant categories following a thorough assessment of diverse approaches and references to multiple survey articles [12,13]. These categories include AES using traditional approaches and AES using neural network approaches. This classification provides a thorough framework for comprehending the many methodologies used in AES systems, bringing clarity and insight into the ever-changing field of assessment technology.

2.1. Traditional AES Approaches Using Hand Crafted Features

Traditional AES systems examine and score essays employing manually crafted features. These systems' performance strongly depend on the selection and quality of these attributes. While successful, manually creating these features takes time, owing to the large amount of available data. Traditional essay scoring approaches rely manually on crafted features with several persistent challenges, typically extracting linguistic and stylistic features like grammar, spelling, sentence structure, and overall readability. However, despite their ability to evaluate specific language features, these approaches struggle with limitations like subjectivity and consistency, high labor intensity, feature engineering complexity, inability to capture deep semantic meaning, and limited adaptability.

A former high school English teacher developed the automated scoring method in the 1960s [14], motivated by the time-consuming chore of manually grading stacks of papers over multiple weekends, and sought help. Despite its inception, the Project Essay Grade (PEG) system was not widely used until the mid-1980s. It was not until the 1990s that PEG had a revival. Ref. [7] suggested a statistical model for automating essay grading that uses data like character count and misspelled words to create an essay vector. Each essay was evaluated using Latent Semantic Analysis (LSA) based on a vector matrix.

The goal of [15] was to create an application for assessing digital English essays using the XGBoost classifier. The system has 12 scoring elements, and uses the datasets of junior high school students' argumentative and narrative essays. Through a 5-fold cross-validation evaluation, the automated essay grader achieves an average accuracy of 66.87%. The Timed Aggregate Perceptron vector model, proposed by [8,16], assesses writings using a combination of factors. This model was trained using word character count, misspelled words, and the n-gram representation. The essay's ranking was used to assess its grade. Furthermore, ref. [9] described a text-mining technique for evaluating short answers. They measured sentence distances to compare the model's and the student's replies.

In a study by [17], Mutual Information Regression was developed to demonstrate the impact of feature selection. The study compared four ML approaches based on linguistic data, and found that the three-layer network with feature selection performed best. To capitalize on the benefits of DL, a novel hybrid model was proposed that combines specific qualities with a higher-level DNN. The hybrid model, as recommended, was more accurate than previous methods. Recent AES breakthroughs overwhelmingly favor English, leaving Chinese AES development behind. Refs. [18,19] provides three specialized machine and DL models for HSK essays that use Word2vec and TF-IDF approaches for feature extraction. Models, including XGBoost and DNN, were examined, with XGBoost utilizing TF-IDF producing the lowest MAE of 6.7%, proving to be the most effective methodology. Whether LSTM or flattened layers, DNN performs sub-optimally in HSK AES [16,20].

Ref. [21] present a novel approach that uses ML and NLP methodologies to automatically assess descriptive answers, including Wordnet, Word2vec, Word Mover's Distance (WMD), cosine similarity, Multinomial Naive Bayes (MNB), and TF-IDF. The results indicate that WMD outperforms cosine similarity in terms of overall performance.

2.2. Neural Network AES Approaches

In recent years, neural network techniques have received considerable attention and recognition in AES [19]. These revolutionary approaches, which fall under the umbrella of AI, have transformed the process of analyzing written compositions. Researchers have investigated fresh approaches to improving AES systems' accuracy, efficiency, and reliability using DL techniques and neural network topologies [20]. This section digs into Neural Network AES Approaches, exploring various approaches and breakthroughs that have transformed the landscape of AES.

2.2.1. Prompt Specific Scoring

Ref. [22] described a DNN-based AES model with a three-layer Recurrent Neural Network (RNN) architecture. The recurrent layer uses the last pooling and bidirectional

LSTM techniques. This framework's innovative component is Score-Specific Word Embedding (SSWE). Comparative evaluations show that SSWE outperforms [23] regarding word embedding quality for essays. Ref. [24] proposed learning vector quantization with a neural network for essay training, allowing the network to assess ungraded essays. Following preprocessing, the framework determines the relevance of the essay content. Previous approaches interpreted essays only in terms of words. However, ref. [25] advocated modeling text hierarchies by developing word and phrase CNNs to build a two-level hierarchical representation approach. Their hierarchical CNN model for AES showed that automatically taught features outperformed handcrafted features in in-domain and domain-adaptation tests. Refs. [26,27] described an attention mechanism, initially proposed by [28], that is suited for a hierarchical representation model and targets certain input data regions. This adaptation allows the attention mechanism neural network architecture to generate accurate scores. Ref. [29] described a short answer scoring engine comprising an ensemble of DNN and a Latent Semantic Analysis-based model designed to evaluate brief constructed responses over a wide range of questions in a national assessment program. Ref. [30] proposed a unique approach that uses item response theory to combine prediction scores from various AES models, considering differences in scoring behavior features between models. The results show that this strategy beats individual AES models and conventional score-integration techniques regarding accuracy. Ref. [31] proposed an AES system that uses coherence and self-attention techniques to compute essay scores while incorporating grammar. On the other hand, ref. [32] proposed combining traditional feature engineering with neural network methods, highlighting their strengths. Building on this idea, ref. [33] proposed a hybrid system that blends handcrafted essay-level characteristics with a DNN AES model, hoping to reap the benefits of both approaches. Ref. [34] used BERT and XLNet to develop a sequence-to-sequence learning model for essay evaluation and grading. Finally, ref. [35] introduced a novel essay scoring method that concurrently trains a multi-scale essay representation using BERT, demonstrating significant improvements over typical approaches employing pretrained language models. Ref. [34] evaluated prompt-specific features using a DNN-AES model. They could effectively predict trait-specific essay outcomes by integrating a hierarchical representational model with an attention mechanism. Wang (2022) described a system for scoring prompt-specific trait assignments and predicting multiple trait scores. This model is based on an RNN architecture designed to produce a variety of outputs. Ref. [36] evaluates the effectiveness of a multi-perspective Hybrid Neural Network (HNN) in assessing student responses in scientific education with an analytic rubric. The study compares the accuracy of the HNN model to four alternative ML approaches: BERT, AACR, Naive Bayes, and Logistic Regression. The findings indicate that the HNN is more accurate than all other methods examined.

2.2.2. Cross Prompt Scoring

Ref. [37] proposed a cross-prompt scoring strategy for AES, using two self-supervised learning tasks per prompt. Refs. [38,39] proposed using transfer learning and DNN to evaluate multi-dimensional writings. A hierarchical fine-tuning method is used to optimize a BERT framework for essay grading. Ref. [40] describes a novel methodology called the Two-stage DNN (TDNN) method, which uses scored essays from non-target prompts to train a scoring model independent of the prompt. This method creates pseudo-ratings for unrated target prompt essays. Furthermore, refs. [41,42] introduces the Shared and Enhanced DNN (SEDNN) framework, which seeks to extract more effective characteristics for essay evaluation. The paper describes a two-stage approach to automated cross-prompt essay grading.

Within the domain of PAES, refs. [43,44] developed a domain-generalized paradigm based on multi-task learning to create a versatile essay scoring system capable of efficiently handling a variety of prompts. Furthermore, forecasting holistic and trait scores emphasized cross-prompt essay trait scoring. This novel architectural design seamlessly combined prompt-specific and trait-specific encodings, improving the model's capacity

to detect complex essay features across various prompts while keeping prompt-specific properties. Furthermore, refs. [13,45] addressed other difficulties related to datasets, characteristics, assessment matrices, numerous methodological strategies, and even the trust and motivation of students on these systems in this field of study. Ref. [46] proposed models based on Recurrent Neural Networks (RNNs) for predicting essay coherence and argument strength, whereas [47] developed a trait scoring model capable of predicting multiple trait scores. Cross-prompt Trait Scorer (CTS) is an innovative approach for simultaneously forecasting holistic and trait scores. Furthermore, refs. [38,48] proposed using transfer learning and DNN for assessing multidimensional essays, with results demonstrated on the CELA dataset.

3. Methodology

This section describes the proposed technique for the research task, which revolves around the complex problem of AES. A robust framework capable of comprehensively evaluating the various components of writing quality is required for accurate and successful essay grading. As a result, the recommended method is intended to automatically assess essays of different genres and topics.

3.1. Dataset Analysis

This study uses the 2012 Automated Student Assessment Prize (ASAP) dataset, which was generated through a cooperation between Kaggle and the Hewlett Foundation. The dataset includes 12,976 essays organized into eight sets, each representing a different genre and essay challenge. Essays include a wide range of themes, including argumentative and narrative essays. The dataset is complex, with word counts ranging from 2 to 1204. These essay lengths challenge the model to understand its context for better prediction. An 80/20 ratio for training and testing is used, respectively.

The dataset analysis for essay scoring reveals key metrics reflecting essay features and their correlation with grades. Minimum and maximum scores of 0 and 60 illustrate the score range. Figure 1 displays essay length distribution, showcasing variations in submission lengths. Examining essay lengths, the analysis considers their average, maximum, and minimum lengths.

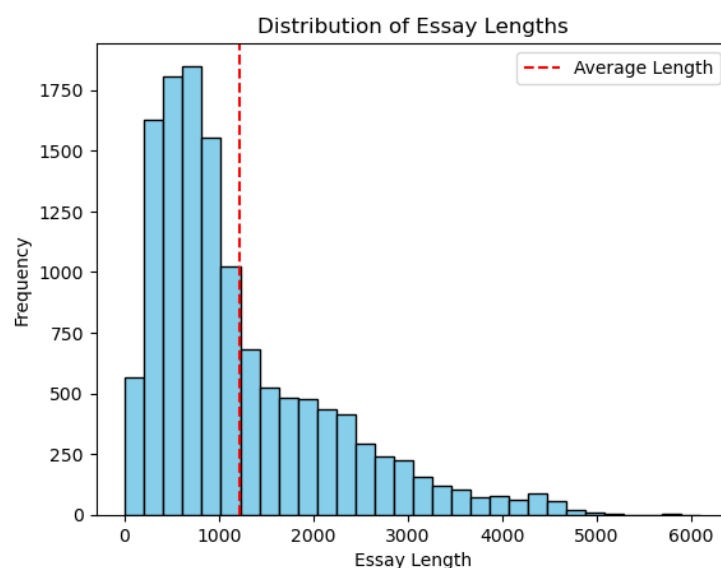


Figure 1. Distribution of essay lengths.

Furthermore, Figure 2 provides a visual representation of the distribution of essay word count, highlighting the diversity in the lengths of the essays. An essay might include as few as two or as many as 1204 words.

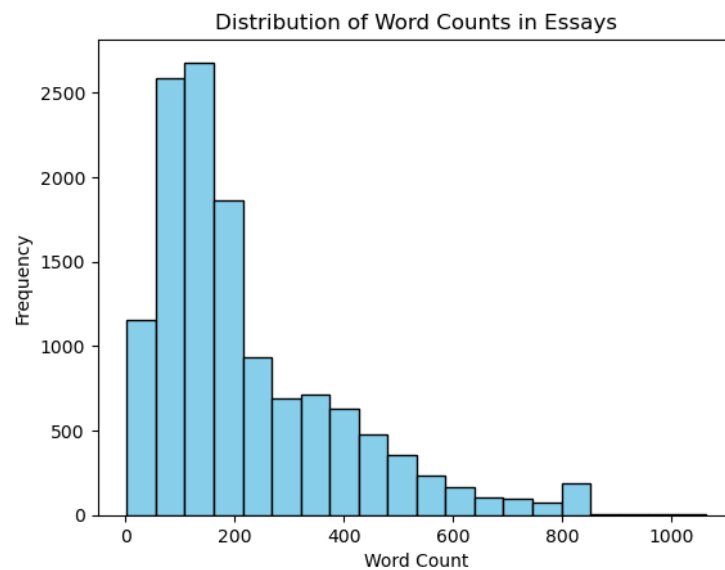


Figure 2. Distribution of essay word counts.

Furthermore, it was discovered that the average number of distinct words in an essay is 119.02, indicating the vocabulary richness throughout the dataset. Descriptive statistics are used to investigate the connection between an essay's word count and its grade. With a median word count of 184, the mean word count is 252.48. The word count's standard deviation is 203.36, which suggests that essay length varies. A correlation coefficient 0.60 indicates a moderate positive correlation between word count and marks received. A boxplot showing the correlation between essay length and grades is shown in Figure 3.

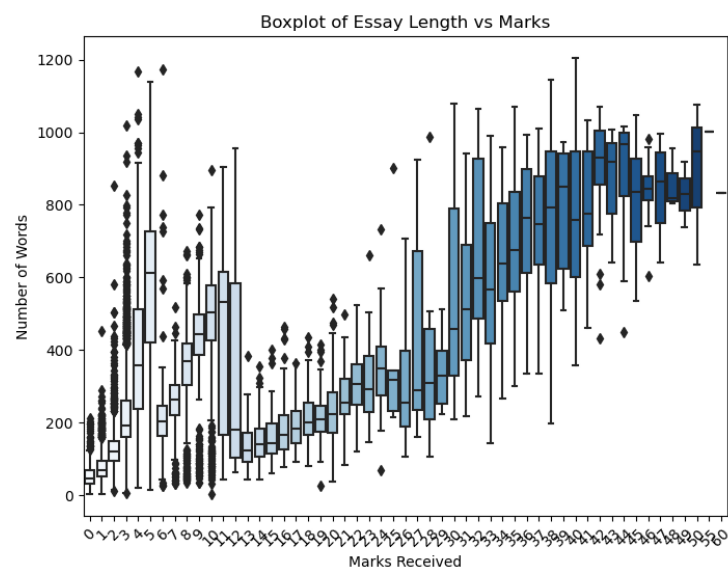


Figure 3. Boxplot of essay length vs. marks.

Additionally, the analysis examines whether getting higher marks for an essay correlates with having a good vocabulary. Vocabulary Richness (TTR) descriptive data show that TTR is 0.53 on average, with a standard deviation of 0.13 and a mean of 0.53. Descriptive data for the marks obtained also show a mean of 6.80, a median of 3, and an 8.97 standard deviation. An understanding of the connection between marks and sentence structure can be gained from looking at Figure 4, which shows the relationship between sentence length and essay score. The results indicate that shorter sentences tend to receive lower marks, while essays with longer, more complex sentences are more likely to achieve high scores.

This highlights the model's sensitivity to sentence complexity as a significant feature in essay evaluation.

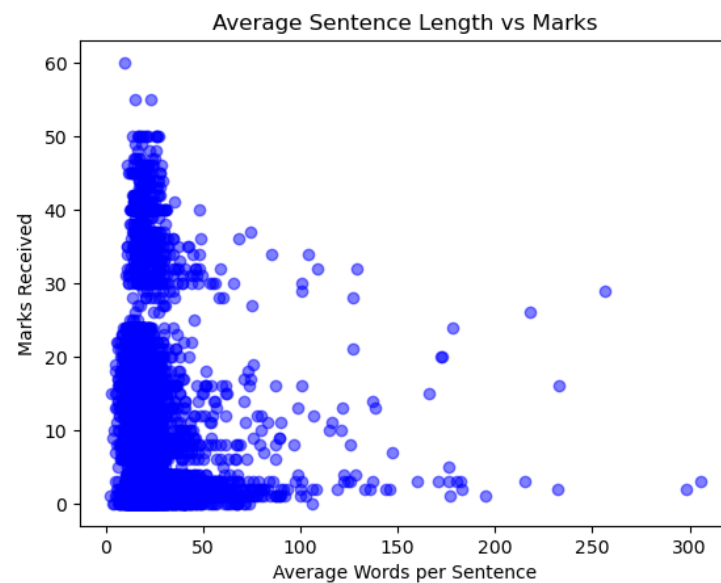


Figure 4. Average sentence length vs. marks.

Last but not least, supplementary figures show the distribution of scores in Figure 5 and the distribution of sentence lengths in the top ten essays in Figure 6. These graphic depictions help to clarify the variables affecting essay scores and provide more information about the dataset's features.

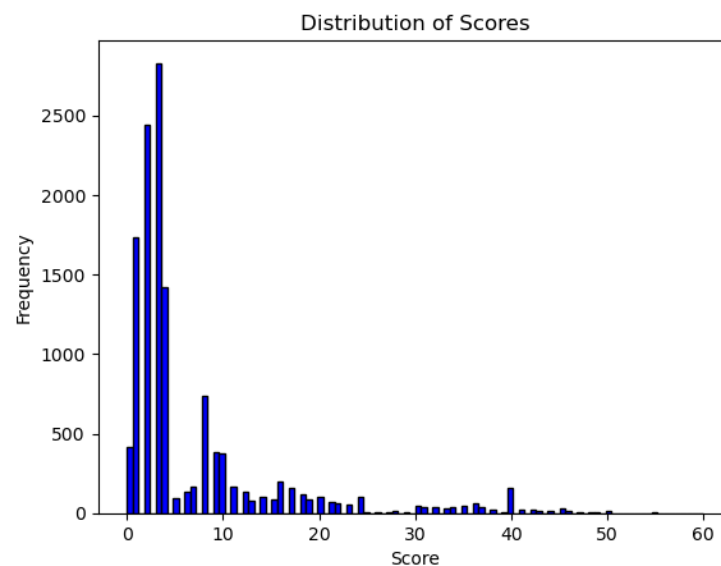


Figure 5. Distribution of scores.

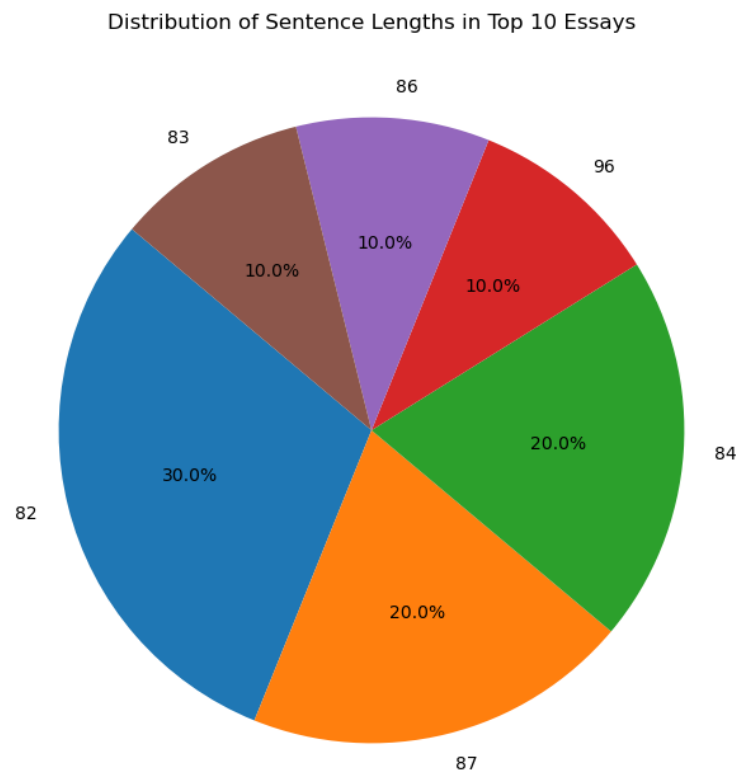


Figure 6. Distribution of sentence lengths in top 10 essays.

3.2. Overview of the Model

The model under consideration combines sophisticated DL methods designed to score essays from the ASAP dataset. The suggested method thoroughly explains essay content and structure by combining handmade features and RoBERTa embeddings. The model is trained to optimize the QWK score, highlighting exceptional performance and accuracy in essay grading, using the LwXGBoost model as the underlying algorithm. For better evaluation, we incorporate MSE and RMSE as well. Using cutting-edge feature extraction techniques and DL, this integrated strategy improves the model's capacity to evaluate essays quickly and accurately. Figure 7 depicts the architecture of the proposed model; refer to Figure 8 for a visual representation of the model's essential components.

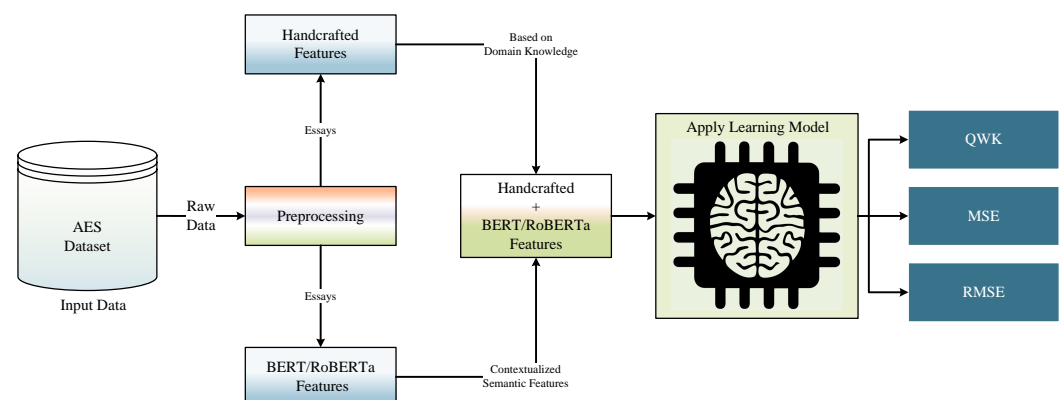


Figure 7. General flow of proposed model.

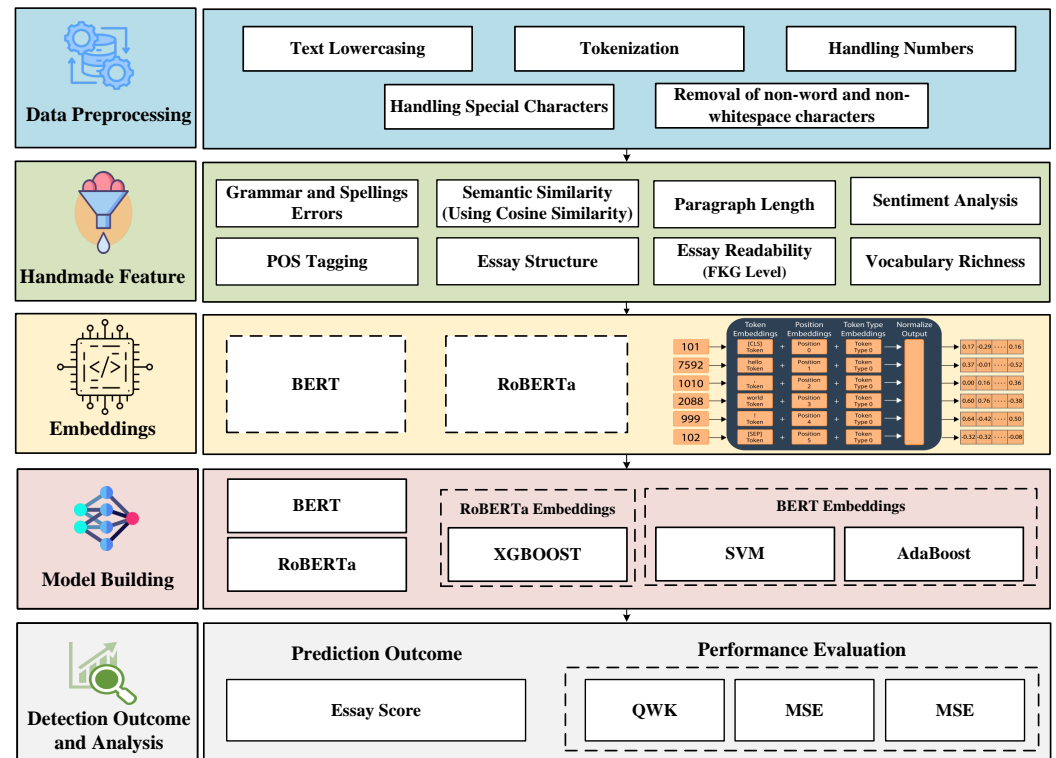


Figure 8. Layered architecture of proposed model.

3.3. Preprocessing

The preprocessing stage is a critical part of the overall pipeline in this work, as it directly influences the quality of the input data and, consequently, the performance of the proposed method [49]. Key operations are applied during preprocessing, such as text lowercasing, tokenization, special characters, handling numbers, and removing non-word and non-whitespace characters. These steps ensure the text data are clean, standardized, and ready for feature extraction and model training.

The preprocessing stage reduces data noise by reducing inconsistencies such as capitalization and needless punctuation, allowing the model to focus on meaningful information rather than extraneous features. Furthermore, tokenization divides the text into individual tokens (words) necessary for subsequent analysis and feature extraction tasks. This makes the dataset more structured and easily analyzed using models such as RoBERTa and LwXGBoost. Finally, preprocessing improves the model's ability to recognize linguistic patterns, increases the quality of retrieved features (both handmade and embeddings), and assures that the input is suitable for ML techniques. Combining cleaner, more consistent input data with enhanced feature extraction results in more accurate and efficient score predictions, contributing to the AES system's overall resilience.

However, if preprocessing is not performed as it should be, it might bias the model, lowering the quality of the extracted features and decreasing the model's predictive accuracy. Errors such as faulty tokenization, failure to appropriately handle special characters, and missing punctuation can distort the text representation and degrade the performance of RoBERTa embeddings and handcrafted features.

Multiple error-handling mechanisms are implemented to address these concerns, including strict validation checks during preprocessing to ensure that only valid tokens proceed to subsequent stages. Missing or faulty data are replaced with placeholders or handled using imputation methods. Additionally, edge cases like numerals or special characters are removed or normalized to provide consistent input for RoBERTa embeddings and other models.

Preprocessing requires multiple processes to ensure that the text is ready for assessment. Lowercasing is the initial step in ensuring consistency across the dataset, converting all text to lowercase. Equation (1) illustrates converting original essays to lowercase. Tokenization, which occurs after lowercasing, divides the text into individual words or tokens. Equation (2) allows for a more detailed analysis of the text's parts. Processing special characters during the preprocessing step of essay scoring is critical to ensuring the text is clear and ready for analysis. This stage involves managing symbols, punctuation marks, and other non-alphanumeric characters that may not directly contribute to the text's semantic meaning. Equation (3) shows how special characters are handled. To remove any remaining non-textual elements from the processed text, non-word and non-whitespace characters must be deleted. This stage simplifies the content by deleting extraneous characters that could impede future analytical processes. Equation (4) illustrates the elimination process.

$$\text{Lowercase}(T_{\text{original}}) = T_{\text{lowercase}} \quad (1)$$

$$\text{Tokenize}(T_{\text{lowercase}}) = \{w_1, w_2, \dots, w_n\} \quad (2)$$

$$\text{RemovePunctuation}(T_{\text{lowercase}}) = T_{\text{no_punctuation}} \quad (3)$$

$$\text{final_text} = \text{processed_text}[\text{replace}(\text{non_word_characters}, ' ')] \quad (4)$$

Then, how numbers are handled is determined by their relevance to the essay's theme. Depending on the situation, numbers can be kept in their current form, converted to words, or removed from the text. Special characters are handled appropriately to maintain text coherence and readability by eliminating or replacing spaces. In conclusion, the text is divided into sentences to enable additional analysis at the sentence level if the analysis calls for such information. Equation (5) represents the segmentation of sentences.

$$\text{SegmentSentences}(T_{\text{no_punctuation}}) = \{s_1, s_2, \dots, s_m\} \quad (5)$$

Algorithm 1 shows the easy-to-understand flow of the preprocessing.

Algorithm 1 Essay preprocessing

Input: Essays

Output: Processed text data

- 1: **procedure** PREPROCESS_ESSAYS
 - 2: **Step 1:** Lowercasing
 - 3: Convert all text to lowercase
 - 4: **Step 2:** Tokenization
 - 5: Split the text into individual words or tokens
 - 6: **Step 3:** Handling Numbers
 - 7: Decide whether to keep numbers as they are, convert them to words, or remove them entirely based on their relevance to the essay's content
 - 8: **Step 4:** Handling Special Characters
 - 9: Handle special characters appropriately, such as deleting them or substituting them with spaces
 - 10: **Step 5:** Removal of Non-Words
 - 11: Remove any non-word characters from the processed text
 - 12: **Return:** Processed text data
 - 13: **end procedure**
-

3.4. Feature Engineering

A key component of AES is feature extraction, which extracts pertinent linguistic and contextual data from the essays to guide the system in calculating scores. With the proposed approach, a thorough portrayal of the writings is obtained using a blend of handcrafted and BERT features. Handcrafted features are essential to AES because they offer insightful information about many linguistic and structural aspects of the essays.

These characteristics improve the overall functionality and interpretability of the scoring system by supplementing the data gathered by neural network models such as RoBERTa.

3.4.1. Grammar and Spelling Errors

These errors are identified using a combination of grammar checking and spacing to count the number of mistakes per essay. The number of grammatical errors is a direct feature that reflects the language accuracy of the essay. If no error is detected, the value will be 0.

3.4.2. Semantic Similarity

We use a pre-trained BERT model to generate embeddings for the essay and a reference text. Then, we calculate the cosine similarity between the two vectors to obtain the semantic similarity score. The maximum score of similarity will be 1. For example, a 0.85 score shows a high similarity index.

3.4.3. Paragraph Length

Analyzing the length of paragraphs provides insights into the essay's organization and coherence, highlighting shifts in focus or development of ideas within the text. We compute the length of paragraphs in the essay by calculating the words.

3.4.4. Sentiment Analysis

Assessing the sentiment expressed in the essays helps gauge the writer's tone, attitude, and emotional engagement, influencing the overall impact and persuasiveness of the arguments presented. We use a TextBlob library to extract sentiment polarity and subjectivity for this feature. We observe a neutral to positive tone for each essay, expressing factual observations with a maximum score of 1. If sentiment polarity is 0.25, it shows a slightly positive sentiment. Similarly, the sentiment subjectivity score of 0.5 indicates a moderately subjective sentiment.

3.4.5. POS Tagging

Employing part-of-speech tagging identifies the essays' grammatical structure and syntactic patterns, facilitating a deeper analysis of sentence construction and linguistic complexity. We count the essay's nouns, verbs, adjectives, and adverbs. We will keep the category values separately.

3.4.6. Essay Structure

Examining the essay's organization and logical flow allows you to assess the effectiveness of the writer's argumentation and the clarity of their message delivery. This rule-based check could assess coherence by looking for transition words such as "however", "therefore", or sentence transitions. We will count how many times each identified word appears.

3.4.7. Essay Readability (Flesch Kincaid Grade Level)

The Flesch Kincaid Grade Level is critical for AES, since it directly affects essay readability, which is essential for effective communication. It aligns with educational goals by adapting writings to readers' reading levels. It also measures clarity and coherence, ensuring that essays are engaging and compelling, making it an essential criterion for judging essay quality and applicability. Calculating the readability score using the Flesch Kincaid Grade Level provides insights into the essays' complexity and accessibility, enabling the assessment of their suitability for the target audience. For example, a score of 7.5 implies that this essay is understandable to a 7th grader.

3.4.8. Vocabulary Richness

Assessing the diversity and sophistication of language usage reveals the writer's linguistic ability and cognitive maturity, which influences the perceived depth and sophistication of the essay topic. To calculate the score, we count the number of unique words and divide by the total amount of words.

Once we have calculated each feature, we will concatenate them column-wise into our original dataset, iterate through each essay, and concatenate the feature scores with the ones already in the dataset. After concatenation, the dataset will be ready for the next step. RoBERTa will extract contextualized embeddings from this new dataset and delicate semantic information from essays. A pre-trained RoBERTa model can efficiently encode complex linguistic patterns and relationships. The proposed model gains a complete understanding of essays by combining handcrafted features with RoBERTa features. Because of this synergy, it can accurately capture structural and semantic qualities, boosting the precision and durability of automated essay assessment.

3.5. Model Building

The model development phase is essential to improving the AES system, since it tests different algorithms and strategies to improve the model's precision and effectiveness in assessing essays in their entirety. The proposed approach thoroughly tested several models in this phase, including BERT, RoBERTa, ILwXGBoost, SVM, and AdaBoost. Each model uses the feature representations retrieved from the preprocessing and feature extraction phases to capture the essays' structural and semantic elements. The details of each model will be covered in more detail in the following sections, along with an explanation of their unique structures and how they enhance the AES system.

3.5.1. BERT

BERT significantly enhances essay scoring by capturing complex contextual relationships and subtle semantic nuances in text. Its bidirectional architecture allows it to understand both the preceding and succeeding words, enabling it to comprehend linguistic patterns and relationships crucial for essay evaluation. This architecture helps BERT to effectively deduce meaning, identify subtle language cues, and ensure coherence in text, which leads to more accurate representations of the essay content.

BERT's pre-training on large-scale textual data enables it to understand diverse language conventions and writing styles. This capability makes it highly versatile and adaptable when scoring essays across different topics and genres. The contextualized embeddings generated by BERT represent the essay's content holistically, which enhances the model's ability to make accurate and insightful predictions about the essay's quality.

The dataset is initially preprocessed before giving to the model, and handcrafted features are extracted and concatenated. BERT embeddings create a robust feature vector for each essay. The final feature vector $\mathbf{x}_i \in \mathbb{R}^{d_1+d_2}$ for each essay e_i is a combination of handcrafted features and BERT embeddings, formulated as

$$\mathbf{x}_i^{\text{hand}} = \phi_{\text{hand}}(e_i), \quad \mathbf{x}_i^{\text{BERT}} = \phi_{\text{BERT}}(e_i), \quad \mathbf{x}_i = [\mathbf{x}_i^{\text{hand}} \mid \mathbf{x}_i^{\text{BERT}}] \quad (6)$$

The combined feature vector \mathbf{x}_i is then used as the input to the scoring model.

For training, the model, parameterized by Θ , uses the combined feature vector \mathbf{x}_i to predict the essay score \hat{y}_i . The QWK score κ is employed to measure the agreement between the predicted scores $\hat{\mathbf{y}}$ and the true scores \mathbf{y} , calculated as

$$\hat{y}_i = g(\mathbf{x}_i; \Theta) \quad (7)$$

The QWK score is formulated as

$$\kappa = 1 - \frac{\sum_{i,j} \omega_{ij} O_{ij}}{\sum_{i,j} \omega_{ij} E_{ij}} \quad (8)$$

where

- O_{ij} is the observed score matrix (confusion matrix between true and predicted scores);
- E_{ij} is the expected score matrix (assuming no agreement beyond chance);
- ω_{ij} is the weight matrix, often defined as $\omega_{ij} = \frac{(i-j)^2}{(k-1)^2}$, where k is the number of possible score levels.

The loss function $\mathcal{L}(\Theta)$, based on the QWK score, is given by

$$\mathcal{L}(\Theta) = 1 - \kappa(\mathbf{y}, g(\mathbf{X}; \Theta)) \quad (9)$$

The objective is to minimize this loss function concerning the model parameters Θ ,

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}(\Theta) \quad (10)$$

Once the model is trained, the score for a new essay e_{new} with its corresponding feature vector \mathbf{x}_{new} is predicted as

$$\hat{y}_{\text{new}} = g(\mathbf{x}_{\text{new}}; \Theta^*) \quad (11)$$

This method provides a comprehensive mechanism for AES by integrating contextualized BERT embeddings with handcrafted features optimized using the QWK score. In our experiment, the following configuration was used to ensure optimal model performance.

The learning rate (2×10^{-5}) and batch size (128) were chosen after tuning using GridSearchCV. These values provided a balance between computational efficiency and model performance. We experimented with batch sizes of 32 and 128, and found that 128 led to the best QWK score and training time performance. Similarly, the learning rate was fine-tuned by experimenting with values ranging from 1×10^{-5} to 5×10^{-5} , with 2×10^{-5} yielding the most consistent results. We also use epochs in the range of 20 to 50. However, the best results were recorded on the 30th epoch (Table 1).

Table 1. Training parameters for BERT model.

Parameter	Value
Number of Epochs	30
Learning Rate (Pretrain)	2×10^{-5}
Learning Rate (Predictor)	1×10^{-2}
Batch Size	128
Maximum Length of Sequences	512
Weight Decay	0.01

3.5.2. RoBERTa

RoBERTa is an improved variant of BERT, designed to overcome some limitations of the original BERT model. Unlike BERT, RoBERTa removes the Next Sentence Prediction (NSP) task, which allows for more efficient learning of sentence-level semantics. RoBERTa is also trained on larger datasets and longer sequences, making it particularly effective in tasks like AES, where understanding local and global context is crucial.

In this work, RoBERTa is a standalone model that directly accepts its embeddings and handcrafted features. These handcrafted features include grammar errors, sentence complexity, and vocabulary richness, complementing RoBERTa's embeddings by adding valuable linguistic structure to improve essay scoring accuracy. The model is explicitly fine-tuned for our AES task without using any external models, such as LwXGBoost.

The final feature vector x_i for each essay e_i is a combination of handcrafted features and RoBERTa embeddings, as shown in Equation (12):

$$x_{\text{hand}}^i = \phi_{\text{hand}}(e_i), \quad x_{\text{RoBERTa}}^i = \phi_{\text{RoBERTa}}(e_i), \quad x_i = [x_{\text{hand}}^i \mid x_{\text{RoBERTa}}^i] \quad (12)$$

This combined vector is fed into the RoBERTa model for training, where it is fine-tuned to optimize essay scoring based on the combined features. The model is trained to optimize three key evaluation metrics: QWK, MSE, and RMSE. These metrics provide a comprehensive view of the model's performance, measuring its agreement with human raters (QWK) and the precision of its predictions (MSE and RMSE).

O_{ij} is the observed score matrix, E_{ij} is the expected score matrix assuming no agreement beyond chance, and w_{ij} is the weight matrix. In addition to QWK, the MSE and RMSE are used to understand the model better. The RoBERTa model is trained to minimize the QWK-based loss function while reducing MSE and RMSE for better accuracy and consistency in essay score prediction (Table 2).

Table 2. Training parameters for RoBERTa model.

Parameter	Value
Number of Epochs	30
Learning Rate (Pretrain)	2×10^{-5}
Learning Rate (Predictor)	1×10^{-2}
Batch Size	64
Maximum Length of Sequences	256
Weight Decay	0.01

The learning rate (2×10^{-5}) and batch size (64) were chosen after tuning using Grid-SearchCV. These values provided a balance between computational efficiency and model performance. We experimented with batch sizes of 32 and 128, and found that 64 led to the best QWK score and training time performance. Similarly, the learning rate was fine-tuned by experimenting with values ranging from 1×10^{-5} to 5×10^{-5} , with 2×10^{-5} yielding the most consistent results. We also use epochs in the range of 20 to 50. However, the best results were recorded on the 30th epoch.

3.5.3. LwXGBoost

LwXGBoost is a powerful and efficient algorithm well-suited for tasks such as essay scoring. It excels in handling high-dimensional data and uncovering intricate essay patterns by leveraging gradient-boosting techniques. Through its ensemble learning framework, which consists of decision trees as base learners, LwXGBoost can model nonlinear relationships and interactions between features, leading to more accurate predictions. LwXGBoost is vital for combining RoBERTa embeddings with handcrafted features because it can handle high-dimensional data and complex feature interactions. We chose RoBERTa to reduce memory and computation load and develop a lightweight model, and we also used 256 tokens to make the model more lightweight for embeddings. LwXGBoost offers computational efficiency by reducing the complexity of constructing decision trees; this study focused on maximizing predictive accuracy and robustness. To develop an LwXGBoost model, we choose a maximum depth of 3 to reduce the complexity and training time. The learning rate was 0.1 for stability. During this process, we prioritized securing the best result without affecting the model performance and making it lightweight.

In this work, LwXGBoost and RoBERTa are synergistically combined to enhance the accuracy and robustness of the essay scoring system. RoBERTa generates contextual embeddings, capturing deep semantic relationships within the essay text. These embeddings allow the model to understand nuanced meanings and coherence across the essay. However, while RoBERTa excels at semantic understanding, it does not fully address important structural features like grammar, sentence length, and readability. LwXGBoost is a robust learning algorithm integrating RoBERTa embeddings and manually engineered features to bridge this gap. By processing this combined feature set, LwXGBoost effectively models the complex interactions between semantic and structural aspects of essays, leading to more precise and resilient score predictions. Integrating RoBERTa deep contextual insights with

LwXGBoost's ability to handle high-dimensional and non-linear data interactions forms a hybrid approach that surpasses the limitations of using either method independently.

This approach uses handcrafted features and RoBERTa embeddings as input to LwXGBoost. Let $E = \{e_1, e_2, \dots, e_n\}$ be a set of n essays. For each essay e_i , a handcrafted feature vector $\mathbf{x}_i^{\text{hand}} \in \mathbb{R}^{d_1}$ is extracted, where d_1 is the dimension of the handcrafted feature space, and a RoBERTa embedding feature vector $\mathbf{x}_i^{\text{RoBERTa}} \in \mathbb{R}^{d_2}$ is also obtained, where d_2 is the dimension of the RoBERTa embedding space,

$$\mathbf{x}_i = [\mathbf{x}_i^{\text{hand}} \mid \mathbf{x}_i^{\text{RoBERTa}}] = [\phi_{\text{hand}}(e_i) \mid \phi_{\text{RoBERTa}}(e_i)] \quad (13)$$

where ϕ_{hand} represents the feature extraction function for handcrafted features and ϕ_{RoBERTa} represents the RoBERTa model used to extract the embeddings.

The combined feature matrix $\mathbf{X} \in \mathbb{R}^{n \times (d_1 + d_2)}$ for all n essays is defined as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} \quad (14)$$

Let $\mathbf{y} = [y_1, y_2, \dots, y_n]^\top$ be the vector of true essay scores. LwXGBoost is trained as a regression model f , which predicts the scores using the combined feature vectors,

$$\hat{y}_i = f(\mathbf{x}_i; \Theta) \quad (15)$$

where Θ represents the parameters of the LwXGBoost model, including the trees' structure and the leaves' weights, the LwXGBoost model is optimized using a custom loss function based on the QWK score, measures the agreement between predicted scores $\hat{\mathbf{y}}$ and true scores \mathbf{y} . We also used MSE and RMSE to evaluate the model performance better.

$$\kappa = 1 - \frac{\sum_{i,j} \omega_{ij} O_{ij}}{\sum_{i,j} \omega_{ij} E_{ij}} \quad (16)$$

where

- O_{ij} is the observed score matrix (confusion matrix between true and predicted scores);
- E_{ij} is the expected score matrix (assuming no agreement beyond chance);
- ω_{ij} is the weight matrix, typically defined as $\omega_{ij} = \frac{(i-j)^2}{(k-1)^2}$, where k is the number of possible score levels.

The objective function guiding the optimization of LwXGBoost is defined as

$$\mathcal{L}(\Theta) = -\kappa(\mathbf{y}, f(\mathbf{X}; \Theta)) \quad (17)$$

Once trained, the model predicts the score for a new essay e_{new} with feature vector \mathbf{x}_{new} as

$$\hat{y}_{\text{new}} = f(\mathbf{x}_{\text{new}}; \Theta) \quad (18)$$

The hyperparameters list, which is used to train the LwXGBoost model and make it more lightweight, follows (Table 3).

For LwXGBoost, we set the maximum depth to 3 to reduce model complexity and training time and avoid overfitting the training data. The learning rate of 0.1 was selected after testing different rates between 0.01 and 0.3, with 0.1 providing the best accuracy and training speed balance. GridSearchCV was used to optimize the number of estimators and other hyperparameters.

Table 3. Hyperparameters for LwXGBoost.

Hyperparameter	Value
n_estimators	50
learning_rate (eta)	0.1
max_depth	3
min_child_weight	1
subsample	1.0
colsample_bytree	1.0
booster	gbtree
eval_metric	None
scale_pos_weight	1
tree_method	auto

3.5.4. Support Vector Machine (SVM) for Essay Scoring

The SVM model is particularly effective in essay scoring because it analyses complex feature spaces and makes accurate predictions. The workflow for essay scoring with SVM begins with the original dataset, followed by the calculation of handcrafted features. These handcrafted features are concatenated with the original feature set, forming a unified dataset.

Next, RoBERTa embeddings are generated from this concatenated dataset, capturing the essays' semantic and contextual information. The combined set of RoBERTa embeddings and handcrafted features is the final input to the SVM model. The SVM model then processes this input to predict essay scores by leveraging the deep contextual representations from RoBERTa and the linguistic features from the handcrafted feature set.

$$\text{Score} = \sum_{i=1}^n \alpha_i y_i K([\mathbf{x}_i^{\text{RoBERTa}} \mid \mathbf{x}_i^{\text{hand}}], \mathbf{x}) + b \quad (19)$$

where

- Score: represents the predicted score of the essay.
- α_i : Lagrange multipliers representing the support vectors.
- y_i : true labels corresponding to the support vectors.
- $K([\mathbf{x}_i^{\text{RoBERTa}} \mid \mathbf{x}_i^{\text{hand}}], \mathbf{x})$: kernel function that computes the similarity between the support vectors $[\mathbf{x}_i^{\text{RoBERTa}} \mid \mathbf{x}_i^{\text{hand}}]$ and the input feature vector \mathbf{x} .
- b : bias term.

Below are hyperparameters for SVM, which are used to train and optimize the model (Table 4).

Table 4. Hyperparameters for SVM.

Hyperparameter	Value
C (Regularization Parameter)	1.0
kernel	rbf
gamma	scale
degree (for polynomial kernel)	3
coef0 (for polynomial/sigmoid kernel)	0.0
shrinking	True
probability	False
tol	1×10^{-3}
max_iter	25

For the SVM model, we chose the radial basis function (RBF) kernel as it performed best in capturing complex relationships within the data. The regularization parameter C was set to 1.0 after testing values ranging from 0.1 to 10.0. This value provided a good balance between margin maximization and classification error. The gamma parameter was

set to ‘scale’, which is based on $1/(n_{\text{features}} \times \text{var})$, as it resulted in more stable training compared to fixed values. Shrinking was enabled to reduce computation time, and the maximum number of iterations was set to 25, which provided an optimal training speed without sacrificing accuracy.

3.5.5. AdaBoost

AdaBoost is a robust algorithm that enhances prediction accuracy and robustness in essay scoring. It works by training weak learners sequentially, where each learner focuses on the aspects of the data that previous learners misclassified. AdaBoost can prioritize previously misclassified instances by assigning more weight to complex samples, thus improving overall performance. This iterative process allows the model to adapt and refine its predictions, leading to higher accuracy. The strength of AdaBoost lies in its ability to combine weak learners into a robust ensemble model, capturing intricate interactions in the data. The following equation represents the scoring process using AdaBoost Equation (20):

$$\text{Score} = \sum_{t=1}^T \alpha_t h_t(x) \quad (20)$$

where

- Score: represents the predicted score of the essay.
- T : number of weak learners (iterations).
- α_t : weight assigned to the predictions of the t^{th} weak learner.
- $h_t(x)$: prediction of the t^{th} weak learner.

Below is a table outlining the default hyperparameters for AdaBoost, which train the model for better performance (Table 5).

Table 5. Hyperparameters for AdaBoost.

Hyperparameter	Value
n_estimators	70
learning_rate	2.0
base_estimator	DecisionTreeClassifier (with max_depth = 1)
algorithm	SAMME.R
random_state	None

For AdaBoost, the number of estimators (70) and the learning rate (2.0) were chosen based on experiments using GridSearchCV. Increasing the number of estimators beyond 70 did not result in significant performance improvement, while it increased the computation time. Similarly, the learning rate was selected after experimenting with values between 0.5 and 3.0, where 2.0 provided the best trade-off between model stability and accuracy. We opted for the DecisionTreeClassifier with a maximum depth of 1 as the base estimator, as it helped reduce overfitting while allowing AdaBoost to focus on difficult-to-classify instances in subsequent iterations.

3.6. Training and Evaluation

Optimizing the performance of the proposed essay scoring model is mainly dependent on the training and evaluation phases. The first step in the process is partitioning the dataset into training and validation subsets with specific portions set aside for model training and evaluation. During the training phase, the model iteratively improves its prediction power by dynamically modifying its parameters in response to the textual attributes taken from the training set. Key metrics like the QWK score, continuously tracked throughout training iterations, provide vital indicators of the model’s success on the training and validation datasets. We also used MSE and RMSE to understand the model performance conveniently. Assessment metrics beyond simple accuracy or loss indicators, such as QWK

score, thoroughly evaluate the model's efficacy in essay scoring. Thus, this comprehensive evaluation approach ensures resilience and dependability in the model's performance for essay-scoring tasks.

4. Experiment Results

This section describes the experimental design and results of the suggested study project. The approach was built Using Pytorch. A maximum of 50 training epochs were conducted, with a five-patient value added to track performance and maybe stop training if performance declined. The dataset was split into an 80/20 ratio for training and validation to guarantee balanced training. This careful data segmentation and the chosen training parameters aim to produce a robust and optimized lightweight model to score essays in the given dataset. Table 6 shows the suggested model's experimental setup.

Table 6. Hardware and software specifications.

Hardware/Software	Description
Operating System	Microsoft Windows 11-x64-based
Physical Memory (RAM)	64 GB
Processor	Intel(R) Core(TM) i9-10900 CPU @ 2.80GHz, 10 Core(s), 20 Logical Processor(s)
Programming Language(s)	Python
IDE	PyCharm Professional

4.1. Evaluation Metrics

Evaluation metrics are essential for evaluating the effectiveness of models in tasks such as essay scoring. They provide numerical metrics for evaluating precision, consistency, and reliability. These criteria are essential for figuring out how well a model works to assess the content and quality of essays when grading them.

QWK is chosen as the primary evaluation metric due to its widespread use in AES to measure agreement with human raters [50]. Because it considers the degree of agreement beyond chance, the QWK score is critical in measuring the agreement between human raters and the model's predictions. Unlike standard accuracy metrics, which only measure the percentage of correctly predicted labels, QWK considers the ordinal nature of essay scores, giving more weight to predictions closer to the true score, thus providing a more nuanced measure of performance. We also report MSE and RMSE to analyze the model performance effectively.

In the proposed approach, QWK is not just a performance metric, but also an integral part of the training process. During model training, QWK is continuously tracked across training epochs to monitor the model's progress in capturing the complexities of essay scoring. The model is optimized to minimize the loss based on QWK, ensuring it is aligned with human grading standards. Furthermore, QWK is used on a test set after training to validate the model's ability to generalize to unseen data. This evaluation method ensures that the model's predictions closely reflect human judgment, making it more reliable for real-world essay-scoring tasks. The QWK score for essay scoring can be calculated using Equation (21),

$$\kappa = 1 - \frac{\sum_{ij} w_{ij} O_{ij}}{\sum_{ij} w_{ij} E_{ij}} \quad (21)$$

where

- κ represents the QWK score.
- O_{ij} denotes the observed agreement between human raters and the model's predictions.
- E_{ij} represents the expected agreement by chance.
- w_{ij} is the weight associated with each pair of ratings.

For calculating MSE and RMSe, we used the following equations:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (22)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (23)$$

where y_i is the true essay score, \hat{y}_i is the predicted score, and n is the total number of essays.

4.2. Results and Evaluation

This study assessed the effectiveness of five ML models for AES: BERT, RoBERTa, LwXGBoost, SVM, and AdaBoost. A wide range of articles from different topics and areas were used to train and evaluate each model. An ensemble technique combining handcrafted features with RoBERTa embeddings improved the model's predictive capabilities.

Model performance was sensitive to hyperparameter choices. For example, increasing the learning rate for the RoBERTa model beyond 2×10^{-5} led to unstable training, whereas lowering it resulted in slower convergence. Similarly, increasing the number of estimators in LwXGBoost beyond 70 resulted in minimal improvement while significantly increasing training time. For AdaBoost, increasing the number of estimators beyond 70 also led to diminishing returns in accuracy. At the same time, for SVM, varying the regularization parameter C outside the range of 0.1 to 10.0 negatively impacted the model's performance. Thus, the chosen hyperparameters were carefully selected to optimize accuracy and efficiency across all models.

With a QWK score of 0.898, an MSE of 0.172, and an RMSE of 0.257, the SVM model could effectively classify essay data and handle high-dimensional feature spaces. SVM performed consistently well in essay scoring assignments because it could identify minute patterns and relationships within the text. This model demonstrates the highest error rate, indicating lower prediction accuracy and agreement with the ground truth.

The AdaBoost model followed with a QWK score of 0.906, an MSE of 0.156, and an RMSE of 0.234. While slightly better than SVM, it still has a higher error rate than more advanced models like BERT and RoBERTa. AdaBoost demonstrated its adaptive boosting technique's capacity to train weak learners sequentially and increase predicted accuracy. AdaBoost's competitive performance in essay scoring can be attributed to its iterative strategy, which allowed it to prioritize complex samples and improve its predictions.

The BERT model achieved a QWK score of 0.918, an MSE of 0.115, and an RMSE of 0.195. This model performs better than SVM and AdaBoost, demonstrating lower error rates and higher predictive accuracy. The BERT model has shown its efficacy in capturing essays' complex contextual details and sophisticated semantics. BERT's precise scoring predictions can be attributed to its bidirectional architecture, which enabled it to understand intricate linguistic structures. An optimized variant of BERT (RoBERTa) also produced a QWK score of 0.927, accompanied by an MSE of 0.101 and an RMSE of 0.179. This model surpassed BERT in performance, showing even lower error rates and a higher agreement score with the target values. The reason for the excellent performance of RoBERTa includes extensive training and dynamic masking, optimized hyperparameters, and training processes.

Finally, LwXGBoost outperformed all other models with the highest QWK score of 0.941, the lowest MSE of 0.071, and an RMSE of 0.1. Its superior accuracy and minimal error rates make LwXGBoost the best performer among the evaluated models. Its ability to identify intricate linkages and patterns in the essay data were improved by integrating several weak learners into a robust ensemble model and the ensemble features. This outstanding result demonstrates how helpful LwXGBoost is for scoring essay tasks. Because of its ability to handle intricate feature interactions and optimize the learning process, LwXGBoost outperforms the other models and produces predictions for essay scores that are more accurate and dependable. Table 7 shows the QWK score obtained by BERT, RoBERTa, SVM, AdaBoost, and LwXGBoost, respectively. The accompanying Figure 9 depicts the LwXGBoost

model's loss curve, visually representing its excellent training dynamics, demonstrating its robust convergence and superior performance over iterations.

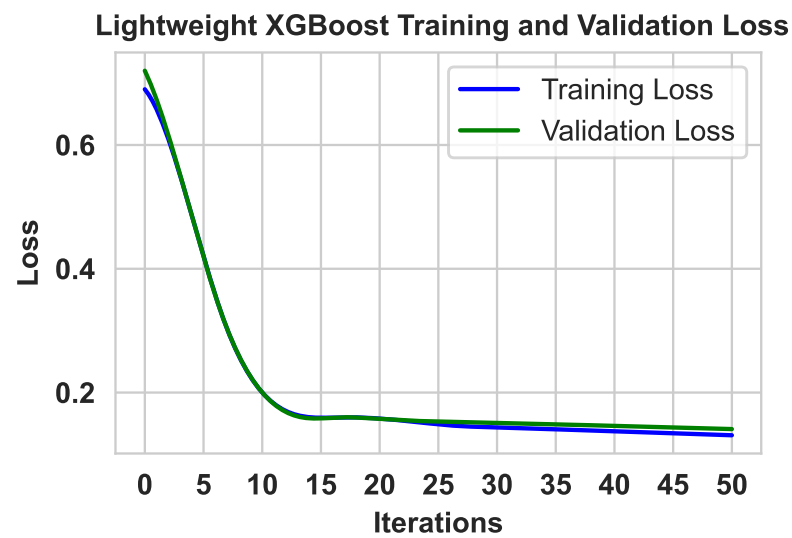


Figure 9. Training and validation loss curve of LwXGBoost.

Figure 10 represents the performance of the distinct models.

Table 7. Model performance metrics.

Model	QWK	MSE	RMSE
SVM	0.898	0.172	0.257
AdaBoost	0.906	0.156	0.234
BERT	0.918	0.115	0.195
RoBERTa	0.927	0.101	0.179
LwXGBoost	0.941	0.071	0.100

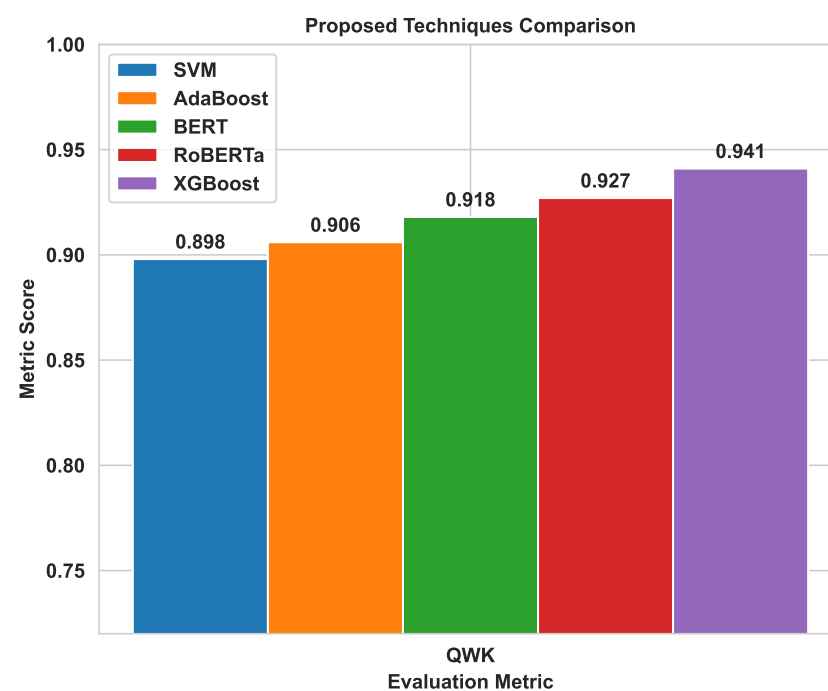


Figure 10. Results of proposed models applied on ASAP dataset.

The suggested method performed noticeably better than other models when compared to them. The proposal in [47] achieved an average QWK holistic score of 0.553 by introducing the Cross-prompt Trait Scorer (CTS). A methodology utilizing DNN and transfer learning, achieving an impressive QWK score of 0.83 on the ASAP dataset, is presented in [38]. Ref. [51] introduced the ProTACT model, demonstrating its competence in holistic scoring with an average QWK holistic score of 0.592. With a QWK score of 0.941, the LwXGBoost model with ensemble features used in this work outperforms these baseline models. Table 8 compares the proposed approach with a different approach. Figure 11 represents the performance of the distinct models with the baseline models.

Table 8. Comparison of approaches.

Model	Average Holistic Score
CTS [47]	0.533
BERT-MTL [38]	0.830
ProTACT [51]	0.592
Proposed Approach	0.941

LwXGBoost was chosen for its superior ability to handle high-dimensional data and model complex feature interactions, particularly when combined with RoBERTa embeddings and handcrafted features. While models like AdaBoost and SVM showed competitive performance, they fell short in crucial areas. AdaBoost struggled with high-dimensional features and could not effectively handle the complexities of semantic embeddings from RoBERTa. Although decisive in separating linear boundaries, SVM could not capture the intricate text relationships in essay data.

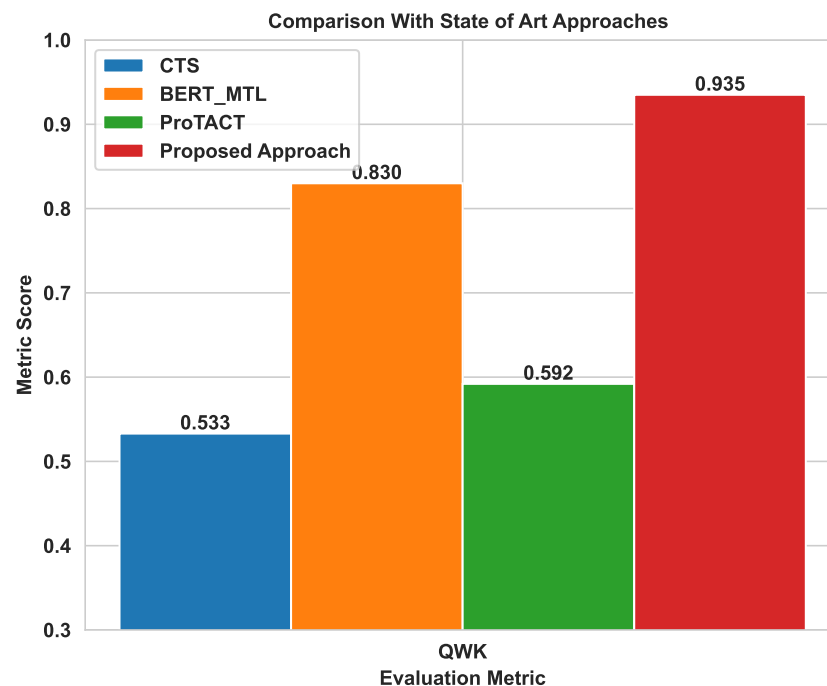


Figure 11. Comparison with state-of-the-art approaches.

LwXGBoost combines gradient boosting with decision trees to identify non-linear patterns and optimize scoring through feature importance and regularization. Its ensemble learning framework, integrating multiple weak learners, ensures it captures both semantic and structural nuances of essays, making it highly effective for AES. Ablation experiments validated this choice, as removing handcrafted features or RoBERTa embeddings led to a significant drop in the model's QWK score, underscoring the importance of both feature types in improving performance.

The model leverages RoBERTa embeddings and handcrafted features to address various essay types and educational levels. RoBERTa embeddings capture nuanced, contextual information within essays, enabling comprehension of complex linguistic patterns regardless of structure. Handcrafted features—such as grammar, sentence structure, and readability—assess more surface-level attributes like coherence and clarity. This combination allows the model to adapt to different essay types, from narrative to argumentative, and across educational levels. For example, essays by younger students with more straightforward language benefit from grammar correction, while more complex essays leverage RoBERTa’s semantic understanding. This hybrid approach ensures robust and accurate scoring across various writing styles and educational levels.

4.3. Comparative Analysis

This study evaluated the performance of various essay scoring models, such as BERT, RoBERTa, SVM, AdaBoost, and LwXGBoost, using the QWK score. The results revealed distinct performance patterns among these models, with LwXGBoost emerging as the top performer, achieving an impressive QWK score of 0.941. Compared to other models, LwXGBoost’s ability to integrate both handcrafted linguistic features and RoBERTa embeddings allows it to capture the complex feature interactions that are often missed by different models, such as RoBERTa or SVM, which either rely purely on embeddings or handcrafted features individually.

- BERT performs exceptionally well at capturing the contextual nuances of text due to its bidirectional transformer architecture, achieving a QWK score of 0.918. However, BERT is limited in handling domain-specific handcrafted features, which can affect its robustness across various prompts and writing styles. On the other hand, RoBERTa showed more good results, having a QWK of 0.927, which shows the effectiveness of RoBERTa on the AES task. We utilized BERT and RoBERTa embeddings with other models such as SVM, AdaBoost, and LwXGBoost. RoBERTa also performs well when embeddings over the BERT are used. The better performance of RoBERTa in our hybrid approach helps capture semantic features, but combining it with handcrafted features further improves the performance of the proposed models.
- SVM, with a QWK score of 0.898, performs well in high-dimensional spaces and is particularly good at separating linear boundaries between different essay scores. However, it cannot leverage deep semantic representations like RoBERTa, and thus struggles to capture more abstract text relationships.
- AdaBoost, scoring 0.906, is an ensemble learning model that focuses on misclassified instances in the data. While it excels at improving predictions for complicated cases, it lacks the flexibility to handle complex, high-dimensional features like RoBERTa embeddings, and its performance falls short compared to LwXGBoost.

By integrating handcrafted features and RoBERTa embeddings within the LwXGBoost model, the proposed hybrid approach achieved superior results with a QWK score of 0.941, effectively capturing semantic and structural features more comprehensively than other models. LwXGBoost handles high-dimensional feature spaces and optimizes complex feature interactions, improving scoring accuracy and resilience to noisy, sparse data and combining contextual embeddings from RoBERTa with traditional handcrafted and original dataset features, showing significant accuracy improvements, particularly for the ASAP dataset. This improvement is further supported by an ablation study, which shows that training with handcrafted features alone yields a QWK score of 0.87. At the same time, RoBERTa embeddings alone result in a QWK score of 0.91. The combined approach, leveraging both feature sets, consistently outperforms these individual components, demonstrating the effectiveness of integrating deep semantic and surface-level linguistic features.

The proposed LwXGBoost enhances existing architectures by utilizing RoBERTa embeddings on a dataset of original and handcrafted features. Previous models relied on DL or traditional ML techniques in isolation, but this study’s hybrid approach leverages both strengths. LwXGBoost outperforms traditional models like Roberta, SVM, AdaBoost,

and advanced models that use only BERT embeddings. Its tree-based structure captures complex interactions between diverse feature sets, managing feature importance and reducing noise through regularization to enhance predictive accuracy. This adaptability makes LwXGBoost particularly effective in leveraging both feature types for essay scoring, resulting in a more comprehensive model than previously achieved with the same dataset. The ablation experiments confirm the importance of both RoBERTa embeddings and handcrafted features in improving the overall performance.

LwXGBoost struggles and faces challenges when dealing with essays that deviate from conventional formats, including those with informal language, non-linear structures, or ambiguous prompts. These essays introduce variability the model finds challenging to interpret, often leading to slight discrepancies in scores compared to human raters. The performance gap from 0.941 to 1 can be attributed to several factors, including the model's sensitivity to language complexity and its inability to fully capture the deeper nuances of content quality, which human graders might prioritize. Additionally, occasional overfitting to training data diminishes generalization to essays with diverse writing styles. Further improvements could enhance the model's ability to handle these more complex and atypical cases, potentially by incorporating additional linguistic features or training on a more varied dataset.

We also explore modern approaches for SOTA comparisons, such as ChatGPT and other advanced LLMs. While models like GPT-4 excel in generating language, they are less specialized for AES tasks than our model, which integrates contextual BERT embeddings and handcrafted linguistic features. Unlike ChatGPT, which focuses on text generation, our model optimizes essay scoring by learning essay-specific features like grammar, readability, and argument structure, ensuring more reliable predictions based on scoring rubrics. The LwXGBoost model captures more intricate feature interactions than traditional LLMs, making it more robust in noisy and sparse data environments. The ablation study further highlights that by combining both handcrafted and deep features, the hybrid approach significantly outperforms models using only one feature set, proving its efficacy in handling diverse essay types and educational levels.

5. Ablation Study

This study evaluates the individual contributions of handcrafted and deep features (RoBERTa embeddings) across different models, demonstrating their impact when used independently and in combination.

5.1. Experimental Configurations

We designed experiments to test each model configuration with the following setups:

- **Handcrafted features only:** the model is trained using only handcrafted features extracted from the essays.
- **RoBERTa embeddings only:** the model is trained using only deep features (BERT embeddings) extracted from the essays.
- **Combined approach:** the model is trained using handcrafted features and RoBERTa embeddings.

Table 9 presents the QWK scores for each model across these configurations.

Table 9. QWK scores for different model configurations.

Model	Handcrafted Features Only	RoBERTa Embeddings Only	Combined Approach
BERT	0.85	0.90	0.918
RoBERTa	0.87	0.89	0.927
SVM	0.82	0.89	0.898
AdaBoost	0.84	0.90	0.906
LwXGBoost	0.87	0.91	0.941

5.2. Observations

The results in Table 9 indicate that while each feature type alone contributes significantly to the model's performance, the highest QWK is achieved when handcrafted and RoBERTa features are combined. Specifically:

- For the **BERT** model, the combined approach improves the QWK score from 0.85 (handcrafted only) and 0.90 (RoBERTa only) to 0.918.
- For the **RoBERTa** model, the combined approach improves the QWK score from 0.87 (handcrafted only) and 0.89 (RoBERTa only) to 0.927.
- For **SVM** and **AdaBoost**, the combined approach also shows performance improvement, although the gains are more moderate compared to the BERT and LwXGBoost models.
- **LwXGBoost**, our top-performing model, achieves the highest QWK score of 0.941 when using the combined features, validating the effectiveness of the hybrid approach.

This ablation study comprehensively compares how each model performs with different feature sets. It highlights the synergy between handcrafted and deep features, showcasing their complementary strengths and validating the benefits of our proposed method.

6. Discussion

This study introduces a new approach that explores the efficacy of advanced techniques in AES. The strategy, which focuses on ensemble learning with LwXGBoost, RoBERTa embeddings, and handcrafted features, outperforms individual models by effectively combining linguistic and semantic features within an ensemble framework. The results underscore the unique value of LwXGBoost in capturing the nuanced complexities of essay content, leading to more accurate scoring outcomes. Furthermore, integrating pre-trained language models like RoBERTa alongside traditional handcrafted features proves crucial in capturing contextualized semantic information, enhancing the model's ability to recognize finer details within essays, and improving scoring accuracy.

While the proposed hybrid approach has significantly improved scoring accuracy, particularly by combining handcrafted features with DL embeddings, limitations still exist. First, the reliance on handcrafted features requires domain expertise, which may not generalize well across diverse topics and prompts. Additionally, while RoBERTa embeddings capture deep semantic information, they do not always address the more subjective aspects of essay scoring, such as creativity or argument strength. Furthermore, the model's performance depends on the dataset, and its adaptability to different essay types, prompts, and genres beyond the ASAP dataset has not been thoroughly tested.

During the error analysis phase, we observed that the proposed model exhibits certain factors that prevent it from reaching a perfect score of 1. It excels in capturing essays' structural and linguistic features but struggles with highly complex or unconventional essay structures that deviate from standard forms. Additionally, human scorers often bring subjectivity to grading, especially in areas like creativity or argument strength, which the model finds challenging to encode. Lastly, noisy data, such as spelling or grammatical errors that escape the preprocessing phase, can lead to less accurate predictions, particularly in sparsely structured essays. These factors collectively contribute to the gap between the model's performance and the ideal.

When comparing the time complexity and scalability of the proposed hybrid approach using LwXGBoost with other methods, LwXGBoost stands out for its efficiency in handling large datasets, utilizing parallel tree boosting, out-of-core computation, and regularization techniques. These features allow LwXGBoost to scale with $O(n \log n)$ time complexity, making it more efficient than models like SVM and AdaBoost, which struggle with high-dimensional data. SVM, in particular, experiences a significant increase in time complexity as data size grows, making it less suitable for large datasets as its time complexity is also $O(n \log n)$. In contrast, while powerful for capturing semantic features, BERT and RoBERTa embeddings have a time complexity of $O(n^2)$ due to the quadratic scaling with input sequence length, resulting in high computational costs for training and inference on large

datasets. However, LwXGBoost efficiently manages the high-dimensional feature space produced by combining RoBERTa embeddings and handcrafted features, leveraging its distributed computing capabilities to maintain scalability even with complex datasets like ASAP. Compared to AdaBoost, which incurs higher computational overhead as weak learners increase, LwXGBoost remains more time-efficient and scalable, particularly in large-scale essay-scoring tasks. Figure 12 shows the best model training time comparison.

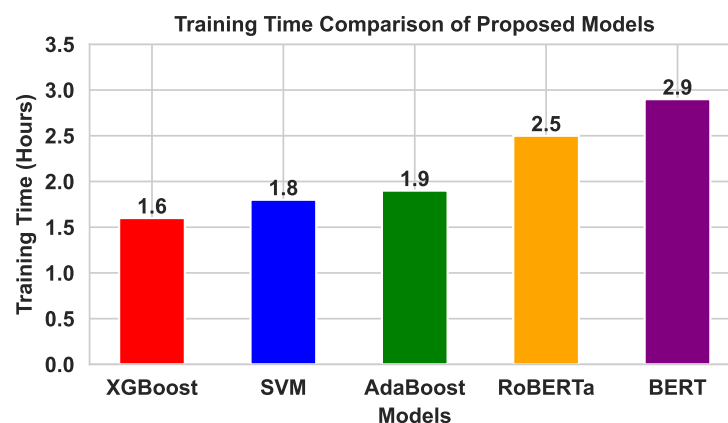


Figure 12. Training time comparison of proposed models.

Despite the LwXGBoost model's high performance, several limitations must be acknowledged. Firstly, although beneficial, reliance on handcrafted features requires substantial domain expertise and may not generalize well across different essay prompts or topics. While LwXGBoost performed exceptionally well on the ASAP dataset, its performance might vary on other datasets with various characteristics, potentially limiting its broader applicability.

7. Conclusions and Future Directions

This study's findings have significant implications for the future of AES. The proposed approach, which combines ensemble learning techniques with BERT embeddings and handcrafted features, significantly enhances the accuracy of essay scoring. These findings lay the groundwork for future advancements in AES systems and underscore the importance of incorporating ethical considerations in their development and deployment. This study's findings are a significant step towards improving AES systems' accuracy and fairness and potentially shaping future research in this field. Other metrics, including MSE, will also be considered for future studies to provide a more comprehensive evaluation. MAE and accuracy are needed to ensure more holistic scoring.

The study also identifies several avenues for future research in AES. These include exploring diverse word embeddings, further refining ensemble learning techniques, and evaluating the proposed approach's applicability across various essay domains. Additionally, the ethical implications of AES systems, particularly the need for fairness and impartiality, are highlighted as critical considerations for their implementation in educational settings. The gravity of these ethical considerations cannot be overstated, and their responsible integration into research and development is essential to ensure that AES systems contribute positively to educational outcomes.

Author Contributions: Conceptualization, M.F. and A.G.; Methodology, M.F. and A.J.; Software, M.F. and A.M.; Validation, M.F., A.J., N.I., A.G., A.A., A.M. and Y.-I.C.; Formal analysis, N.I., A.A. and A.M.; Investigation, A.G.; Resources, A.J., N.I., A.A. and A.M.; Data curation, A.J. and N.I.; Writing—original draft, M.F.; Writing—review & editing, A.G. and Y.-I.C.; Visualization, A.A. and A.M.; Supervision, A.G.; Project administration, Y.-I.C.; Funding acquisition, Y.-I.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Korea Agency for Technology and Standards in 2022. The project numbers are 1415181629 (Development of International Standard Technologies based on AI Model Lightweighting Technologies), 1415180835 (Development of International Standard Technologies based on AI Learning and Inference Technologies), and 1415181638 (Establishment of standardization basis for BCI and AI Interoperability). Any correspondence related to this paper should be addressed to Young-Im Cho.

Data Availability Statement: The original data presented in the study are openly available in Kaggle named “The Hewlett Foundation: Automated Essay Scoring” at <https://www.kaggle.com/competitions/asap-aes> (15 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Annotation	Expanded Form
AES	Automated Essay Scoring
AI	Artificial Intelligence
NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
DNN	Deep Neural Networks
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	Robustly Optimized BERT Pretraining Approach
SVM	Support Vector Machine
AdaBoost	Adaptive Boosting
XGBoost	Extreme Gradient Boosting
Lw XGBoost	Lightweight Extreme Gradient Boosting
ASAP	Automated Student Assessment Prize
QWK	Quadratic Weighted Kappa
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error

References

1. Mizumoto, A.; Eguchi, M. Exploring the potential of using an AI language model for automated essay scoring. *Res. Methods Appl. Linguist.* **2023**, *2*, 100050. [CrossRef]
2. Machicao, J.C. Higher education challenge characterization to implement automated essay scoring model for universities with a current traditional learning evaluation system. In *Proceedings of the International Conference on Information Technology & Systems*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 835–844.
3. Beseiso, M.; Alzubi, O.A.; Rashaideh, H. A novel automated essay scoring approach for reliable higher educational assessments. *J. Comput. High. Educ.* **2021**, *33*, 727–746. [CrossRef]
4. Beseiso, M.; Alzahrani, S. An empirical analysis of BERT embedding for automated essay scoring. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
5. Rahayu, R.; Sugiarto, B. Automated Essay Scoring Using Natural Language Processing And Text Mining Method. In *Proceedings of the 2020 14th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Bandung, Indonesia, 4–5 November 2020; pp. 1–4.
6. Eid, S.M.; Wanas, N.M. Automated essay scoring linguistic feature: Comparative study. In *Proceedings of the 2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*, Alexandria, Egypt, 5–8 November 2017; pp. 212–217.
7. Adamson, A.; Lamb, A.; Ma, R. Automated Essay Grading. In *Proceedings of the Conference on Artificial Intelligence in Education*, Québec City, QC, Canada, 27–31 July 2014.
8. Cummins, R.; Zhang, M.; Briscoe, T. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, 7–12 August 2016.
9. Süzen, N.; Gorban, A.N.; Levesley, J.; Mirkes, E.M. Automatic short answer grading and feedback using text mining methods. *Procedia Comput. Sci.* **2020**, *169*, 726–743. [CrossRef]
10. Liu, Z.; Tang, Q.; Ouyang, F.; Long, T.; Liu, S. Profiling students’ learning engagement in MOOC discussions to identify learning achievement: An automated configurational approach. *Comput. Educ.* **2024**, *219*, 105109. [CrossRef]

11. He, S.; Luo, H.; Jiang, W.; Jiang, X.; Ding, H. VGSG: Vision-Guided Semantic-Group Network for Text-Based Person Search. *IEEE Trans. Image Process.* **2023**, *33*, 163–176. [\[CrossRef\]](#)
12. Uto, M. A review of deep-neural automated essay scoring models. *Behaviormetrika* **2021**, *48*, 459–484. [\[CrossRef\]](#)
13. Ramesh, D.; Sanampudi, S.K. An automated essay scoring systems: A systematic literature review. *Artif. Intell. Rev.* **2022**, *55*, 2495–2527. [\[CrossRef\]](#)
14. Page, E.B. The imminence of... grading essays by computer. *Phi Delta Kappan* **1966**, *47*, 238–243.
15. Salim, Y.; Stevanus, V.; Barlian, E.; Sari, A.C.; Suhartono, D. Automated English digital essay grader using machine learning. In Proceedings of the 2019 IEEE International Conference on Engineering, Technology and Education (TALE), Yogyakarta, Indonesia, 10–13 December 2019; pp. 1–6.
16. Song, W.; Wang, X.; Zheng, S.; Li, S.; Hao, A.; Hou, X. TalkingStyle: Personalized Speech-Driven 3D Facial Animation with Style Preservation. *IEEE Trans. Vis. Comput. Graph.* **2024**, early access. [\[CrossRef\]](#)
17. Birla, N.; Jain, M.K.; Panwar, A. Automated assessment of subjective assignments: A hybrid approach. *Expert Syst. Appl.* **2022**, *203*, 117315. [\[CrossRef\]](#)
18. Xiao, R.; Guo, W.; Zhang, Y.; Ma, X.; Jiang, J. Machine learning-based automated essay scoring system for Chinese proficiency test (HSK). In Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, Seoul, Republic of Korea, 18–20 December 2020; pp. 18–23.
19. Liu, Z.; Xiong, X.; Li, Y.; Yu, Y.; Lu, J.; Zhang, S.; Xiong, F. HyGloadAttack: Hard-label black-box textual adversarial attacks via hybrid optimization. *Neural Netw.* **2024**, *178*, 106461. [\[CrossRef\]](#)
20. Jiang, B.; Zhao, Y.; Dong, J.; Hu, J. Analysis of the influence of trust in opposing opinions: An inclusiveness-degree based Signed Deffuant–Weisbush model. *Inf. Fusion* **2024**, *104*, 102173. [\[CrossRef\]](#)
21. Bashir, M.F.; Arshad, H.; Javed, A.R.; Kryvinska, N.; Band, S.S. Subjective answers evaluation using machine learning and natural language processing. *IEEE Access* **2021**, *9*, 158972–158983. [\[CrossRef\]](#)
22. Alikaniotis, D.; Yannakoudakis, H.; Rei, M. Automatic text scoring using neural networks. *arXiv* **2016**, arXiv:1606.04289.
23. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
24. Shehab, A.; Elhoseny, M.; Hassanien, A.E. A hybrid scheme for automated essay grading based on LVQ and NLP techniques. In Proceedings of the 2016 12th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 28–29 December 2016; pp. 65–70.
25. Dong, F.; Zhang, Y. Automatic features for essay scoring—an empirical study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1072–1077.
26. Dong, F.; Zhang, Y.; Yang, J. Attention-based recurrent convolutional neural network for automatic essay scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 153–162.
27. Shen, J.; Sheng, H.; Wang, S.; Cong, R.; Yang, D.; Zhang, Y. Blockchain-based distributed multi-agent reinforcement learning for collaborative multi-object tracking framework. *IEEE Trans. Comput.* **2023**, *73*, 778–788. [\[CrossRef\]](#)
28. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
29. Ormerod, C.; Lottridge, S.; Harris, A.E.; Patel, M.; van Wamelen, P.; Kodeswaran, B.; Woolf, S.; Young, M. Automated short answer scoring using an ensemble of neural networks and latent semantic analysis classifiers. *Int. J. Artif. Intell. Educ.* **2023**, *33*, 467–496. [\[CrossRef\]](#)
30. Uto, M.; Aomi, I.; Tsutsumi, E.; Ueno, M. Integration of prediction scores from various automated essay scoring models using item response theory. *IEEE Trans. Learn. Technol.* **2023**, *16*, 983–1000. [\[CrossRef\]](#)
31. Li, X.; Chen, M.; Nie, J.; Liu, Z.; Feng, Z.; Cai, Y. Coherence-based automated essay scoring using self-attention. In Proceedings of the Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018, Changsha, China, 19–21 October 2018; pp. 386–397.
32. Uto, M.; Xie, Y.; Ueno, M. Neural automated essay scoring incorporating handcrafted features. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6077–6088.
33. Dasgupta, T.; Naskar, A.; Dey, L.; Saha, R. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, Melbourne, Australia, 19 July 2018; pp. 93–102.
34. Rodriguez, P.U.; Jafari, A.; Ormerod, C.M. Language models and automated essay scoring. *arXiv* **2019**, arXiv:1909.09482.
35. Wang, Y.; Wang, C.; Li, R.; Lin, H. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. *arXiv* **2022**, arXiv:2205.03835.
36. Latif, E.; Zhai, X. Automatic Scoring of Students’ Science Writing Using Hybrid Neural Network. *arXiv* **2023**, arXiv:2312.03752.
37. Cao, Y.; Jin, H.; Wan, X.; Yu, Z. Domain-adaptive neural automated essay scoring. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25 July 2020; pp. 1011–1020.
38. Xue, J.; Tang, X.; Zheng, L. A hierarchical BERT-based transfer learning approach for multi-dimensional essay scoring. *IEEE Access* **2021**, *9*, 125403–125415. [\[CrossRef\]](#)

39. Zhou, Z.; Zhou, X.; Qi, H.; Li, N.; Mi, C. Near miss prediction in commercial aviation through a combined model of grey neural network. *Expert Syst. Appl.* **2024**, *255*, 124690. [\[CrossRef\]](#)
40. Jin, C.; He, B.; Hui, K.; Sun, L. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1088–1097.
41. Li, X.; Chen, M.; Nie, J.Y. SEDNN: Shared and enhanced deep neural network model for cross-prompt automated essay scoring. *Knowl.-Based Syst.* **2020**, *210*, 106491. [\[CrossRef\]](#)
42. Qiao, G.; Hou, S.; Huang, X.; Jia, Q. Inclusive tourism: Applying critical approach to a Web of Science bibliometric review. *Tour. Rev.* **2024**. [\[CrossRef\]](#)
43. Ridley, R.; He, L.; Dai, X.; Huang, S.; Chen, J. Prompt agnostic essay scorer: A domain generalization approach to cross-prompt automated essay scoring. *arXiv* **2020**, arXiv:2008.01441.
44. Ding, J.; Chen, X.; Lu, P.; Yang, Z.; Li, X.; Du, Y. DialogueINAB: An interaction neural network based on attitudes and behaviors of interlocutors for dialogue emotion recognition. *J. Supercomput.* **2023**, *79*, 20481–20514. [\[CrossRef\]](#)
45. Conijn, R.; Kahr, P.; Snijders, C.C. The effects of explanations in automated essay scoring systems on student trust and motivation. *J. Learn. Anal.* **2023**, *10*, 37–53. [\[CrossRef\]](#)
46. Mim, F.S.; Inoue, N.; Reiser, P.; Ouchi, H.; Inui, K. Unsupervised learning of discourse-aware text representation for essay scoring. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; pp. 378–385.
47. Ridley, R.; He, L.; Dai, X.y.; Huang, S.; Chen, J. Automated cross-prompt scoring of essay traits. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 13745–137530 [\[CrossRef\]](#)
48. Gu, X.; Chen, X.; Lu, P.; Lan, X.; Li, X.; Du, Y. SiMaLSTM-SNP: Novel semantic relatedness learning model preserving both Siamese networks and membrane computing. *J. Supercomput.* **2024**, *80*, 3382–3411. [\[CrossRef\]](#)
49. Faseeh, M.; Khan, M.A.; Iqbal, N.; Qayyum, F.; Mehmood, A.; Kim, J. Enhancing User Experience on Q&A Platforms: Measuring Text Similarity based on Hybrid CNN-LSTM Model for Efficient Duplicate Question Detection. *IEEE Access* **2024**, *12*, 34512–34526.
50. Doewes, A.; Kurdhi, N.; Saxena, A. Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring. In Proceedings of the 16th International Conference on Educational Data Mining, EDM 2023, Bengaluru, India, 11–14 July 2023; pp. 103–113.
51. Do, H.; Kim, Y.; Lee, G.G. Prompt-and trait relation-aware cross-prompt essay trait scoring. *arXiv* **2023**, arXiv:2305.16826.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.