A Hybrid Syntactic–Statistical–Semantic Framework for Detecting AI-Generated Text Across Domains

Doaa Mostafa, Sally S. Ismail, and Mostafa Aref

***Abstract*—Recent advances in large language models (LLMs) have enabled highly human-like text generation, raising concerns related to misinformation, authorship verification, and academic integrity. Current approaches for detecting LLM-generated text suffer from several limitations, including limited robustness to linguistic diversity, sensitivity to text length variations and paraphrasing, weak domain generalization, and high computational cost. To address these challenges, this paper proposes a hybrid framework for detecting LLM-generated text that integrates syntactic and statistical features with deep semantic representations learned using GloVe embeddings, Convolutional Neural Networks (CNNs), and Bidirectional Long Short-Term Memory (BiLSTM) networks. By combining linguistic cues with contextual semantics, the proposed model captures both structural and semantic patterns to distinguish human-written text from LLM-generated content. Experiments conducted on the ChatGPT Research Abstracts and ElectAI datasets demonstrate strong cross-domain generalization and robustness to text length variations and paraphrasing. The proposed framework achieves an accuracy of 98.63%, an F1-score of 98.66%, and a minimum false positive rate (FPR) of 0.01. These results indicate the effectiveness, stability, and reliability of the framework for detecting LLM-generated text.**

***Index Terms*—Large language models (LLMs), AI-generated text, Text generation, Word Embedding, Feature Extraction.**

I. INTRODUCTION

Recent advances in natural language generation (NLG) have significantly improved the fluency, coherence, and diversity of the text produced by LLMs. Advanced generative models such as GPT-4 [1], Claude [2], and Gemini [3] now generate writing that is becoming sufficient to achieve human-level performance across tasks such as question answering, email drafting, news article composition, scientific writing, story generation, and code generation. However, alongside these capabilities, significant concerns regarding the potential misuse in areas such as phishing [4], misinformation [5], and academic integrity [6-8]. Current approaches for identifying LLM-generated text generally fall into four categories: watermarking, feature-based, neural-based, and human-assisted approaches [9]. Watermarking approaches embed hidden signals into text during the generation process; however, they require full access to the underlying model, making them impractical for most modern LLMs. Also, watermarking methods are sensitive to minor text modifications, such as adding space, substituting smaller words with larger words, using similar words, which can significantly reduce detection accuracy [10]. Feature-based approaches rely on linguistic indicators such as syntax, grammar, and other stylistic features. They can also suffer degradation in effectiveness when applied to advanced LLMs[10]. Neural-based approaches use pre-trained transformer models [11,12] to classify LLM-generated content. Although these methods achieve strong performance, they require substantial computational resources and large annotated datasets for training [9,10]. Finally, human-assisted approaches rely on human judgment, however, as LLM-generated text becomes increasingly indistinguishable from human writing, reliable manual detection becomes progressively more challenging [10].

To address the limitations of existing detection approaches, this paper proposes a hybrid framework for identifying LLM-generated content that combines handcrafted syntactic, statistical features and deep semantic representations derived from GloVe [13], CNNs [14], and BiLSTM networks together. By using a hybrid framework with interpretable linguistic features and high-level context-based semantics, the proposed framework achieves improved reliability and generalizability across multiple domains and diverse LLMs.

The structure of this paper is as follows. Section 2 reviews the most recent literature on detecting text generated by LLMs. Section 3 presents the proposed framework. Section 4 describes the datasets, evaluation metrics, experimental results, and the discussion of results. Finally, Section 5 will conclude the paper and provide potential avenues for future research.

II. RELATED WORK

This section provides background on the most recent research on detecting LLM-generated text. Recent approaches to detecting LLM-generated text can be broadly categorized into watermarking-based, neural-based, and feature-based detection methods.

Doaa Mostafa, Sally. S. Saad, and Mostafa Aref are affiliated with Computer Science Department, Faculty of Computer and Information science, Ainshams University, Cairo, Egypt. (E-mails: doaa.ahmed74@yahoo.com, sallysaad@cis.asu.edu.eg, Mostafa.aref@cis.asu.edu.eg).

DOI: 10.36244/ICJ.2026.1.6

Watermarking Framework for LLMs (WLLM) [15] embeds imperceptible signals into generated text by biasing token selection during generation. Although WLLM is lightweight and efficient in controlled environments, it essentially depends on access to the generation process and is extremely susceptible to paraphrasing or post-editing, which can greatly weak or eliminate the watermark [9,16]. This limits its applicability in real-world, open-text scenarios.

REMARK-LLM [17] increases the robustness of watermarks by embedding identifiers into internal semantic representations and decoding them via a retrieval-based mechanism. Although this enhances resistance to mild paraphrasing, the approach incurs higher computational costs and still degrades under stronger paraphrasing attacks. Moreover, it requires specialized retrieval and decoding components, reducing scalability.

In DNA-GPT [18], genetic signatures are incorporated in the text generation process in the form of marks, which supports traceability. In spite of achieving a high level of detection accuracy, this model is difficult to implement and requires access to the generation pipeline. The model is also affected by fluctuations in text length.

Giant Language Model Test Room (GLTR) [19] in order to identify LLM-generated text by analyzing token-level statistics like likelihood, rank, and entropy. Although GLTR is interpretable and efficient for older language models, it is less reliable for advanced LLM-generated text, edited text, and paraphrased text and requires access to model probability outputs. AuthentiGPT [20] uses a feature-based, multi-stage detection approach that integrates watermarking signals, semantic embeddings, and linguistic cues. Despite its effectiveness, the method's practicality for large-scale or real-time deployment is limited by the need for substantial computational resources and large labeled datasets.

SeqXGPT[21] is a sentence-level approach that employs sequential models and contextual embeddings to identify linguistic and semantic inconsistencies between human-written and LLM-generated text. This approach generalizes well across various LLM types due to its use of sentence-based architecture. However, the approach is computationally expensive, as processing sentences individually increases resource demands, and its accuracy may drop when sentences are edited or when stylistic cues become harder to distinguish.

In summary, despite recent advances in current detection approaches, they face important restrictions, including dependence on access to the source generation model, prohibitive computation cost, and weak generalizability across models and domains. Many existing detectors also struggle with paraphrased input and short text segments. These issues demonstrate the need for more generalizable, interpretable, and text- level detection. The proposed framework, which utilizes a combination of crafted linguistic and statistical features alongside deep semantic representations, addresses these gaps in performance and detection mode through robust and reliable detection performance.

III. PROPOSED ARCHITECTURE

The proposed framework for detecting LLM-generated text consists of six stages: (1) Preprocessing, in which the input text is cleaned by removing irrelevant elements. (2) Handcrafted statistical and syntactic feature extraction, in which statistical and syntactic features are computed from pre-processed text. (3) Text representation using GloVe embeddings, which convert each token into a fixed-length dense vector that captures semantic relationships through word co-occurrence patterns. (4) Semantic feature extraction, where a CNN captures local contextual patterns and a BiLSTM models long-range dependencies. (5) Feature fusion, in which handcrafted statistical and syntactic features are concatenated with the semantic features produced by the CNN and BiLSTM to form a unified feature representation. (6) Classification was then performed by taking the combined feature vector and passing it through a fully connected layer to finally predict the label. The architecture of the proposed framework is illustrated in Figure 1.

A. Phases of the Proposed Framework

1) Text Preprocessing

The goal of preprocessing is to remove task-irrelevant content (e.g., URLs, emails, symbols, hashtags, numbers) that do not contribute to the linguistic, grammatical, or semantic features used in this study [22]. Such tokens were found to have minimal impact on detection performance and may introduce noise, particularly in short or informal texts, as the MFAD framework primarily relies on stylistic and semantic cues to identify AI-generated text. Therefore, preprocessing uses lowercasing, normalization, lemmatization, and tokenization to get the text ready for trustworthy feature extraction.

2) Statistical and Syntactic Features Extraction

Statistical and syntactic features capture the underlying structure of the text and reflect key indicators such as readability and writing style, which serve as strong cues for assessing textual originality and coherence. Lexical diversity is quantified through measures of vocabulary richness. The statistical features represent measurable aspects of the textual structure, style, and complexity. Part-of-speech (POS) tag frequency, and bigram frequency are used to measure syntactic complexity, which expose syntactic and lexical patterns indicative of human-written versus LLM-generated text [23]. A detailed list of the statistical and syntactic features used in this study is provided in Table I.

3) Text Representation

The final step of text preprocessing is tokenization, which the text is split into individual tokens. These tokens are then passed to GloVe, which models both global statistical relationships and local contextual meanings among words in the dataset. GloVe represents each word as a dense vector in continuous space, typically with 50, 100, 200 or 300 dimensions, where semantically similar words are positioned closer together.

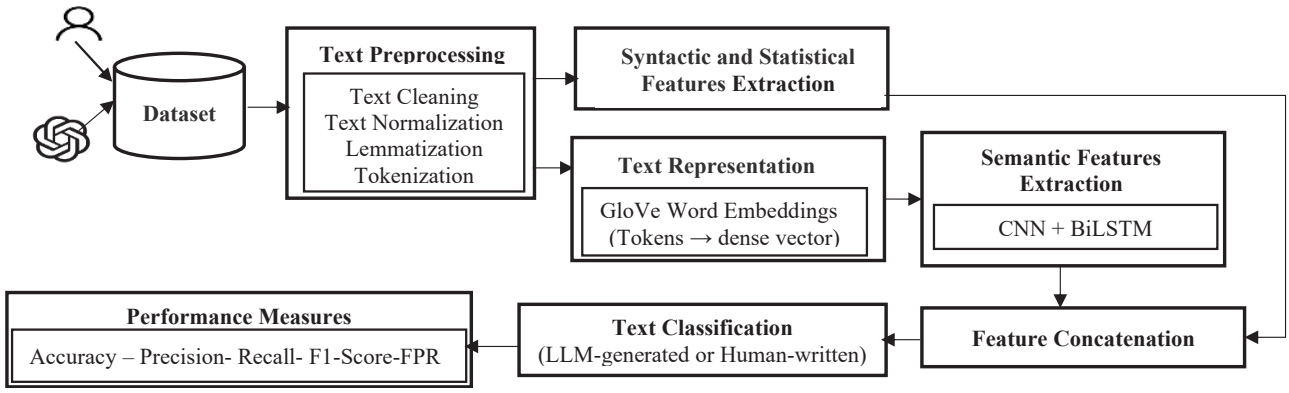


Fig. 1. Architecture of the proposed framework

These embeddings, derived from word co-occurrence statistics in a large corpus, enable GloVe to model semantic relationships very well. In contrast to context-dependent models such as BERT, GloVe embeddings are context-independent in nature, and the model assigns every word the same vector without any additional context-based meaning, preserving semantic proximity based on global co-occurrences. The numerical vectors generated by GloVe are then passed to the CNN for higher-level feature extraction and semantic representation learning. The embedding matrix is constructed according to equations (1–3).

Given a sequence of T tokens:

$$X = [w_1, w_2, \dots, w_T] \quad (1)$$

Each token w_t is mapped to its GloVe embedding:

$$\mathbf{e}_t = \text{GloVe}(w_t) \in \mathbb{R}^d \quad (2)$$

The embedding matrix is constructed as follows:

$$E = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_T]^\top \in \mathbb{R}^{T \times d} \quad (3)$$

where T is the length of the input sequence, w_t is the token at position (t), et is the GloVe embedding vector for the token, d is the embedding dimensionality, and E is the sequence embedding matrix.

4) Semantic Feature Extraction

High-level semantic features were extracted using a combination of CNN and BiLSTM architectures. The CNN consists of convolutional and pooling layer. The convolutional layer learns patterns within the local context and the pooling layer reduces the dimensionality of the features for computational efficiency and robustness. As convolutional filters slide across the text, the CNN captures local n-grams and phrase-level contextual representations, with max pooling selecting the most informative activations from each filter. The obtained local feature maps are then passed into a Bidirectional Long Short-Term Memory (BiLSTM) layer to model long-range dependencies by processing the sequence in both forward and backward directions. This allows each word to be interpreted within its full context. By integrating CNN for capturing local patterns and BiLSTM for modeling global semantic connections into one structure, the framework produces a rich and comprehensive contextual representation.

5) Feature Concatenation

The numerical vectors representing syntactic and statistical features were fused with the semantic feature vectors generated by the CNN+BiLSTM model to form a unified feature vector for each text sample. These feature types provide complementary information: syntactic and statistical features measure structural and quantifiable properties of writing, and the semantic features describe meaning and contextual relations between words. Together, these features yield a richer and more comprehensive representation of the text, thereby enhancing the model's ability to distinguish human-written content from LLM-generated text.

6) Text Classification

Subsequently, the concatenated feature vector is fed into a classifier implemented using the TensorFlow framework. The classifier consists of a fully connected (dense) neural network with a sigmoid activation function in the output layer. This activation function maps the network output to a value between 0 and 1, corresponding to the probability of being in either of two classes, which is well suited for binary classification tasks, predicting whether a text is written by a human or generated using an LLM.

B. Rationale for Model Architecture.

We adopt a GloVe + CNN + BiLSTM architecture rather than fine-tuned Transformer-based models like BERT and DistilBERT for several reasons. Primarily, our aim is to propose an efficient and interpretable detection mechanism that can generalize across domains and LLMs without invoking the need to perform heavy fine-tuning. Transformer-based detectors often involve substantial computational cost involved in training and inference processes of transformer models, as well as sensitivity to domain-specific data distributions.

In contrast, GloVe provides stable corpus-level semantic representations. CNNs can also adequately represent local n-gram information, and BiLSTMs can represent long-range dependencies. This provides effective modeling of semantics but also with a lower computational cost. Additionally, separating handcrafted features from deep semantic

representations allows clearer attribution of performance gains, which supports more interpretable analysis and ablation studies

TABLE I
STATISTICAL AND SYNTACTIC FEATURES

Type	Feature	Description
Statistical [23]	Total Words	The total count of words.
	Total Sentences	The total number of sentences.
	Total Unique Words	The total count of unique words.
	Type-Token Ratio	TTP evaluates vocabulary diversity.
	Total Stop Words	The total count of stop words.
	Total Punctuation	The total count of punctuation marks (commas, periods, exclamation marks, etc.).
	Total Discourse Markers	The total number of discourse markers.
	Total Spelling Errors	The total count of spelling errors.
	Total Grammar Errors:	The total count of grammar errors.
	Readability Scores	The ease of reading the text was evaluated; higher scores indicated easier readability.
	Syllable Count	Total number of syllables.
	Average-Sentence Length	The average number of words per sentence.
	Average word length	The average number of characters per word.
Syntactic [23]	Part-of-Speech (POS) Tag Distributions	Frequencies of noun singular (NN), noun plural (NNS), verb base-form (VB), verb past tense (VBD), adjectives (JJ), adverbs (RB), personal pronoun (PPR), preposition (IN), verb present participle (VBG), contracting conjunction (CC), and determiners (DT).
	Complexity of Sentence	Tree depth: The depth of the syntactic tree reflects sentence complexity, such as the average dependency tree, depth, maximum of dependency tree depth, and number of subordinate clauses.
	Top Bigram/Trigram Frequency	Calculate the maximum number of two- or three-word combinations.

IV. RESULTS

A. Datasets

The ChatGPT Research Abstracts [24] and ElectAI [25] datasets were used for the experiments. The ChatGPT Research Abstracts dataset contains 10,000 titles for papers with a combination of Human and ChatGPT-generated (GPT 3.5) abstracts for each title, and can be used to differentiate between human and AI-generated text. The ElectAI dataset is a collection of English tweets that refer to elections and political claims, tagged as human-written or AI generated, and contains approximately 9,400 tweets. The dataset has been produced using numerous LLMs including Llama-2-7B [26], Mistral-7B [27] and Falcon-7B [28]. Table II provides an overview of the dataset metadata and composition.

TABLE II
METADATA AND DATASET COMPOSITION OF THE CHATGPT RESEARCH ABSTRACTS AND ELECTAI DATASETS.

Dataset	Generation Model	Domian	LLM generated articles Count	Real Data Count (Human)
ChatGPT Research Abstracts	GPT 3.5	Scientific Writing	10000	10000
ElectAI	Llama-2-7B	Political Tweets	2350	2350
	Mistral-7B		2350	
	Falcon-7B		2350	

B. Experimental Setup

Each experiment was conducted on a Lenovo laptop equipped with 12 GB of RAM, an Intel Core i5 processor, and a 64-bit operating system. The proposed framework was implemented in Python 3.10.5 using several libraries: scikit-learn [29] for computing evaluation metrics, NLTK [30] for preprocessing, such as tokenization, stemming, lemmatization, and stop-word removal, and TensorFlow [31] for constructing and training deep neural networks. Additional NLP libraries were used to extract the statistical and syntactic features listed in Table I. In particular, advanced syntactic analyses, including syntactic complexity and dependency-based features, were performed using spaCy [32]. Text complexity and readability metrics were computed with TextStat [33]. Grammatical errors were detected using LanguageTool [34]. The number of spelling mistakes in the text is calculated using PySpellChecker [35]. The hyperparameter configuration is provided in Table III, and includes batch size, number of epochs, optimizer, and dropout rate.

C. Performance Measures

Five classification metrics were used to evaluate the performance of the proposed framework: accuracy, precision, recall, F1-score, and false positive rate (FPR) [9]. Accuracy measures the proportion of texts that are classified correctly, as shown in Eq. (4). Precision represents the proportion of texts classified as AI-generated, that are truly AI-generated, as shown in Eq. (5). Recall (True Positive Rate) measures the proportion of AI-generated texts that were correctly classified as AI-generated, as defined in Eq. (6). The F1 score is the harmonic mean of precision and recall and defined as a uniform indicator of both recall and precision, as given in Eq. (7). Finally, the FPR quantifies the proportion of human-written texts that are incorrectly classified as AI-generated, as defined in Eq. (8).

$$\text{Accuracy} = \frac{(\text{Tp}+\text{TN})}{(\text{TP}+\text{FP}+\text{TN}+\text{FN})} \quad (4)$$

$$\text{Precision} = \frac{\text{Tp}}{(\text{TP}+\text{FP})} \quad (5)$$

$$\text{Recall} = \frac{\text{Tp}}{(\text{TP}+\text{FN})} \quad (6)$$

$$\text{F1-score} = 2\times\frac{(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})} \quad (7)$$

$$\text{FPR} = \frac{\text{FP}}{(\text{FP}+\text{TN})} \quad (8)$$

Here, TP (True Positive) denotes to the number of LLM-generated texts that are correctly classified as LLM-generated. TN (True Negative) represents the number of human-written texts that are correctly classified as human-written. FP (False Positive) refers to human-written texts that are incorrectly classified as LLM-generated and FN (False Negative) refers to LLM-generated texts that have been incorrectly classified as human written.

TABLE III
THE HYPERPARAMETER SETTINGS

Parameters	Value
Activation Function (feature extraction)	ReLU
Activation function (classification)	Sigmoid
Optimizer	Adam
Loss Function	binary_crossentropy
Batch size	32
No. of. Epochs	5
Dropout-rate	0.5
Glove Embedding dimensions	100
Pooling Type	Max
Kernel-Size	3
Dataset splits	70% → training, 10→ validation, and 20% → testing

D. Experimental Results

This section presents the performance of the proposed framework on the ChatGPT Research Abstracts and ElectAI datasets, including detection performance across different dataset categories, robustness to varying text lengths, resilience to paraphrasing, and comparison with baseline approaches.

1) Evaluation of the Proposed Framework on the Two Datasets

The proposed framework is evaluated on the ChatGPT Research Abstracts and ElectAI datasets, as illustrated in Figures 2–3.

These figures illustrate the effectiveness of the proposed framework in distinguishing human-written text from LLM-generated text on both the ChatGPT Research Abstracts and ElectAI datasets. In the ChatGPT Research Abstracts dataset, the model achieves 96.63% accuracy with exceptionally high precision (96.52%) and recall (97.61%), resulting in a strong F1-score of 97.06% and an exceptionally low false positive rate (FPR) of .02. In the ElectAI dataset, the results indicate the model's efficacy across various generation models. In the Human vs Falcon category, the model reaches 96.65% accuracy with 95.60% precision and 97.53% recall. Notably, this performance increases when detecting text generated by Mistral (98.3%) and LlaMA (98.63%). These results confirm the framework's ability to generalize across domains and various LLMs, along with reasonable levels of precision and excellent recall rates, and very few false positives.

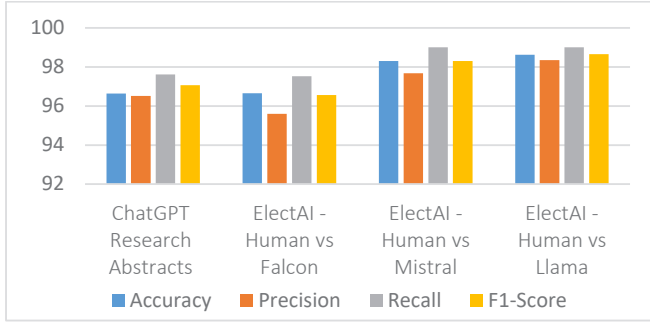


Fig. 2. Evaluation metrics of the proposed framework on the ChatGPT Research Abstracts and ElectAI datasets.

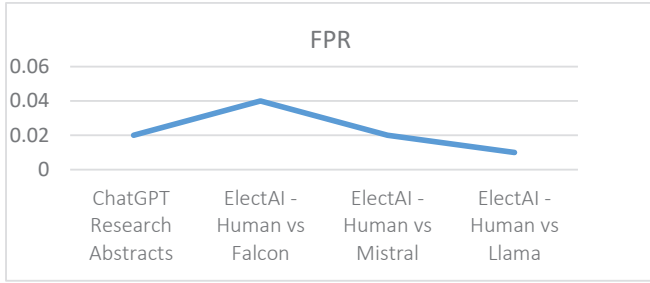
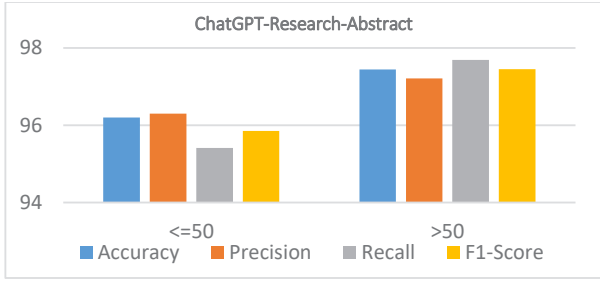


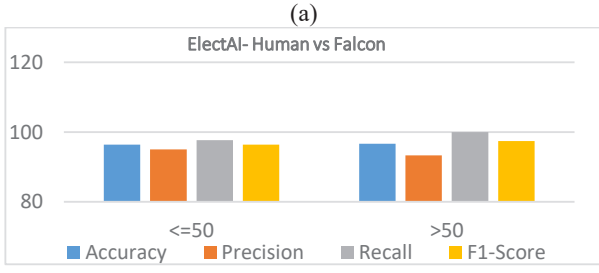
Fig. 3. FPR of the proposed framework on the ChatGPT Research Abstracts and ElectAI datasets.

2) Evaluation of the Proposed Framework on the Two Datasets Across Different Text Lengths

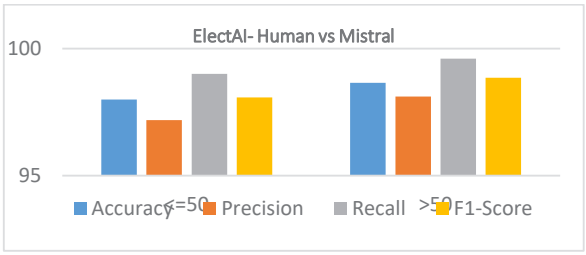
In this section, the proposed framework is evaluated under varying text lengths by partitioning each dataset into two token-count categories: texts with fewer than 50 tokens and texts with more than 50 tokens. The corresponding evaluation results are presented in Figure 4.



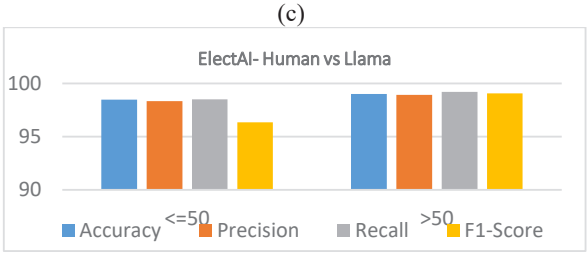
(a)



(b)



(c)



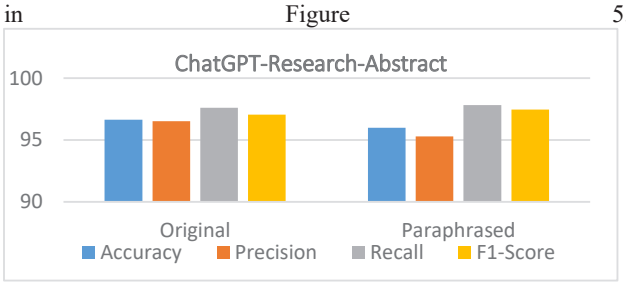
(d)

Fig. 4. Evaluation metrics of the proposed framework on ChatGPT Research Abstracts and ElectAI datasets for different text lengths

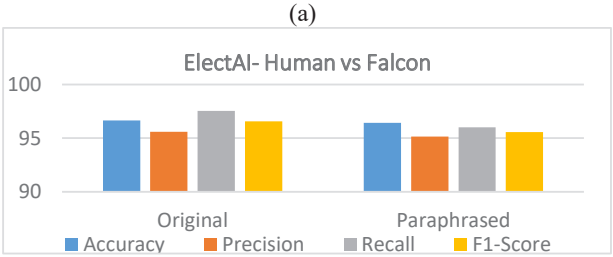
Figure 4 illustrates the evaluation of the proposed framework under varying text lengths on the ChatGPT Research Abstracts and ElectAI datasets supports the underlying robustness and stability of the framework, regardless of the text input length. On the ChatGPT Research Abstracts dataset, the model achieves strong performance with both short (≤ 50 words) and longer (> 50 words) texts. Notably, accuracy increased from 96.20% with short texts to 97.44% for longer texts. This trend continues throughout the ElectAI dataset across all generation models. In the Human vs Falcon classification, the model achieves strong accuracy for both short (96.42%) and long (96.67%) sample versions, with 100% recall being provided for the longer text. Additionally, performance improved among Mistral and LlaMA classifications, with both short and long text versions producing very high accuracy, precision, recall and F1-score measures. Across all text lengths, the results remained above 98% for almost every metric. This indicates that the proposed method is highly effective regardless of text size, consistently achieving strong precision and recall while generalizing well across different AI models and text-length variations.

3) Evaluation of the Proposed Framework on the Two Datasets under Text Paraphrasing

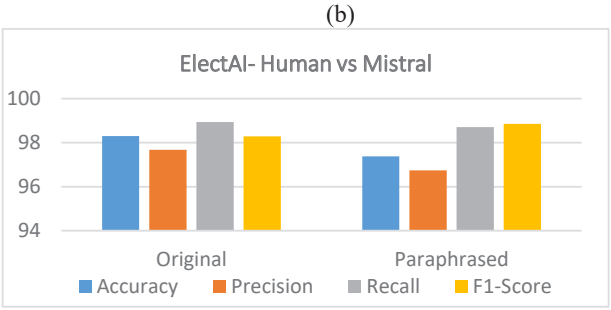
The proposed framework was evaluated on its ability to detect paraphrased LLM-generated texts that preserve semantic meaning while introducing syntactic variation. Paraphrased samples were generated using a pre-trained T5-base transformer model, which follows a text-to-text learning paradigm [36]. The model employs an encoder–decoder architecture, where the decoder creates a rephrased sentence based on the contextualized semantic representation produced by the encoder. The generated paraphrases introduce lexical and syntactic variation, including word substitutions, reordering, and structural reformulation, without changing the underlying meaning, T5's paraphrased outputs are intended to be semantically equivalent to the original text. This allows the evaluation of the robustness of the proposed framework under realistic paraphrasing-based text transformations. After the paraphrasing process, the proposed framework was applied to the modified dataset, and its performance was compared with that obtained on the original, non-paraphrased test set. The evaluation results under paraphrasing conditions are presented in Figure 5.



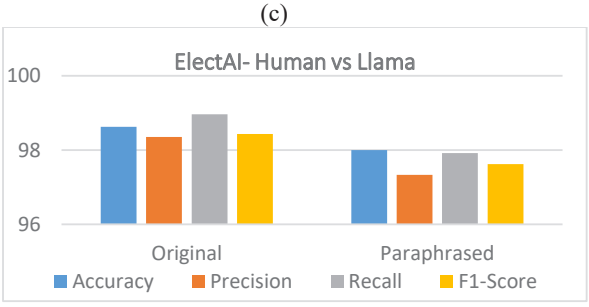
(a)



(b)



(c)



(d)

Fig. 5. Evaluation Metrics of the Proposed framework Between Original and Paraphrased Texts in the ChatGPT Research Abstracts and ElectAI Datasets.

Figure 5 illustrates the evaluation of the proposed framework under paraphrasing conditions and shows that the model remains highly effective when confronted with syntactically altered but semantically equivalent text. On the ChatGPT Research Abstracts dataset, performance declines only marginally, with accuracy decreasing from 96.63% on the original texts to 96% on paraphrased versions. Meanwhile, recall exhibits a slight increase, indicating sensitivity to AI-generated patterns despite syntactic variation. A similar trend is

observed on the ElectAI dataset. In the Human vs. Falcon category, performance remains stable, with only a minimal decrease in accuracy and F1-score after paraphrasing. Even for more advanced models such as Mistral and LLaMA, the framework continues to perform strongly, achieving accuracies of 97.38% and 98% on paraphrased texts, respectively, which are close to the results obtained on the original data. Although minor reductions in precision and F1-score are observed, overall detection performance remains robust. These results demonstrate that the proposed framework generalizes well to paraphrased LLM-generated content and effectively preserves deeper semantic and stylistic cues even after substantial rewriting.

4) Comparison of the Proposed Framework with Baseline Approaches

This section compares the proposed framework with representative baseline approaches. As shown in Table IV, the proposed framework consistently outperforms AuthentiGPT and SeqXGPT on both the ChatGPT Research Abstracts and ElectAI datasets.

TABLE IV
PERFORMANCE ANALYSIS OF THE PROPOSED FRAMEWORK IN COMPARISON WITH BASELINE APPROACHES

Dataset	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ChatGPT Research Abstracts	AuthentiGPT	88	84	87	85.5
	SeqXGPT	85.9	85.5	90.5	88
	Proposed	96.6	96.5	97.6	97
ElectAI (All)	AuthentiGPT	92	86.3	90.7	88.5
	SeqXGPT	86.4	82.3	93.3	87.5
	Proposed	97.8	97.2	98.5	98

The results in Table IV show that the proposed framework outperforms AuthentiGPT and SeqXGPT on both the ChatGPT Research Abstracts and ElectAI datasets. On the ChatGPT Research Abstracts dataset, the proposed framework achieves an accuracy of 96.6%, outperforming AuthentiGPT by 8.6 percentage points and SeqXGPT by 10.7 percentage points. Precision, recall, and F1-score all show comparable improvements. Notably, the proposed framework achieves a 97% F1-score, which is much higher than that of SeqXGPT (88%), and AuthentiGPT (85.5%). SeqXGPT's exhibits relatively high recall (90.5%), indicating sensitivity to AI-generated content; however, its lower precision and overall accuracy suggest a higher false-positive rate. On the other hand, the proposed framework effectively distinguishes between text produced by AI and text written by humans while maintaining high recall and precision (97.6%, 96.5%). A similar trend is observed on the ElectAI dataset. The proposed framework achieves an accuracy of 97.8%, outperforming AuthentiGPT (92%) and SeqXGPT (86.4%). It also records the highest precision (97.2%) and recall (98.5%), resulting in an F1-score of 98%. These results highlight the proposed framework's strong generalization capability across different domains and writing styles. While AuthentiGPT performs competitively on this dataset, its lower precision and F1-score suggest limitations in capturing nuanced stylistic and semantic differences. SeqXGPT again demonstrates relatively high recall (93.3%) but lower precision (82.3%), reinforcing the trade-off between sensitivity and specificity observed in feature-based baselines.

5) Ablation Study

To evaluate the contribution of each component in the proposed framework, an ablation study will be carried out by considering different feature types, sequences modeling architectures, and word embeddings on the ChatGPT Research Abstracts and ElectAI datasets. First, we evaluate handcrafted features syntactic and statistical stylometric indicators, to measure their individual contribution. Next, we evaluate semantic features extracted using CNN combined with sequence models such as RNN, LSTM, and BiLSTM, with GloVe and Word2Vec embeddings. We then examine the impact of feature fusion by combining handcrafted features with semantic features, where CNN and RNN are used with GloVe embeddings. Finally, the reference model combines handcrafted features with semantic features, which are learned with GloVe, CNN, and BiLSTM techniques, to serve as a reference model to evaluate all the individual metrics.

The results in Table V show that handcrafted syntactic and statistical cues are insufficient for reliable AI-generated text detection, as handcrafted features alone only achieve relatively low accuracy on both ChatGPT Research Abstracts (69.8%) and ElectAI (60.7%). Semantic-only configurations substantially improve performance, consistently exceeding 90% accuracy, highlighting the importance of deep semantic representations. The benefit of bidirectional contextual modeling is confirmed by the fact that using a standard RNN in place of BiLSTM causes a discernible drop in performance, while LSTM improves results but is still marginally less effective than the full proposed framework. Using Word2Vec embeddings further enhances semantic-only performance. The combined framework, integrating semantic and handcrafted features, achieves the highest accuracy and F1-score, demonstrating the complementary strengths of contextual and stylometric patterns.

E. Discussion

The experimental results confirmed the effectiveness and robustness of the proposed hybrid method for identifying LLM-generated. The most significant result from this study shows how important it is to combine different types of features (syntactic, statistical, and deep semantic) to achieve a reliable and generalized performance when detecting LLM-generated text. The addition of statistical and syntactical features accurately captures the surface structure and stylistic characteristics of LLM-generated text and the subtle differences between the writing styles of LLMs and humans. At the same time, using CNNs and BiLSTMs provides the model with a deeper understanding of the semantics of the text, enabling the model to detect more subtle differences in linguistic behaviour and the coherent use of word meaning. This framework shows strong performance (i.e., accuracy, precision, recall) as well as a low false-positive rate across both the scientific abstracts and political tweets datasets.

Generalization across different LLMs (GPT-3.5, LLaMA-2,

Mistral, Falcon) indicates robustness across varying datasets and styles of generation. The framework is highly consistent despite changing lengths of text, both less than and more than 50 tokens in length. Also, this framework maintains strong performance when classifying items that had been syntactically modified (paraphrased) using the T5 Model. For instance, paraphrasing the ChatGPT Research Abstracts dataset with T5 resulted in only a minor accuracy drop from 96.63% to 96%. The proposed framework consistently showed superior performance across all datasets and LLM categories When compared to the feature-based baseline AuthentiGPT and SeqXGPT. Therefore, the combination of multiple feature types is an effective way of increasing the sensitivity and reliability of an LLM text detector. Finally, the paraphrasing evaluation focuses on controlled, single-step neural rewriting to simulate common automated rewriting attacks. Future work will investigate more challenging cases, including multi-step paraphrasing, back-translation, and human-edited paraphrases, to further validate the framework's robustness.

TABLE V
ABLATION STUDY RESULTS OF THE PROPOSED FRAMEWORK

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ChatGPT Research Abstracts	Handcrafted Features only	69.8	68.8	72.7	70.7
	Semantic Features only (Glove+ CNN+BiLSTM)	95.6	95	96.8	95.89
	Semantic Features only (Glove+ CNN+LSTM)	94.8	94.4	92.8	93.6
	Handcrafted+ Semantic (Glove+ CNN+RNN)	92.6	93	92.3	92.6
	Semantic Features only (Word2Vec+ CNN+BiLSTM)	94.6	93.9	96	95
	Proposed Framework	96.6	96.5	97.6	97
ElectAI (All)	Handcrafted Features only	60.7	62.1	59.2	60.6
	Semantic Features only (Glove+ CNN+BiLSTM)	96.2	96.5	97.6	97
	Semantic Features only (Glove+ CNN+LSTM)	94.6	95.7	96.6	96
	Handcrafted+ Semantic (Glove+ CNN+RNN)	93.7	92	92.8	93
	Semantic Features only (Word2Vec+ CNN+BiLSTM)	95.22	94.6	96.8	95.7
	Proposed Framework	97.8	97.2	98.5	98

V. CONCLUSION

This paper presents a hybrid detection framework that combines syntactic and statistical features with deep semantic representations derived from CNN and BiLSTM models. The proposed framework is effective in differentiating between human-written text and LLM-generated text across multiple domains and generative models by leveraging complementary feature types. Experiments on scientific abstracts and political tweet demonstrate high performance, with the framework achieving a maximum accuracy of 98.63%, an F1-score of 98.66%, and a minimum false positive rate (FPR) of 0.01. This framework also demonstrates strong robustness to variations in text length and strong resilience in detecting paraphrased LLM-generated content. These results indicate the advantage of fusing different feature types to improve LLM-generated text detection. Future work will involve extending the use of this framework with larger and more diverse multilingual datasets and evaluating its performance on data generated from more advanced LLMs. Additionally, this framework will be enhanced to improve adversarial robustness, such as real-world adversarial paraphrasing, and provide a more comprehensive comparative assessment with current detection approaches.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, and D. Almeida, "Gpt-4 technical report," *arXiv preprint; arXiv:2303.08774*, 2023, DOI: 10.48550/arXiv.2303.08774.

[2] A. Priyanshu, Z. Hong, and Y. Maurya, "AI Governance and Accountability: An Analysis of Anthropic's Claude," *arXiv preprint; arXiv:2407.01557*, 2024, DOI: 10.48550/arXiv.2407.01557.

[3] G. Team, R. Anil, J. Yu, R. Soricut, S. Borgeaud, J. B. Alayrac, and J. Schalkwyk, "Gemini: a family of highly capable multimodal models," *arXiv preprintar; arXiv:2312.11805*, 2023, DOI: 10.48550/arXiv.2312.11805.

[4] A. Giaretta and N. Dragoni, "Community targeted phishing, "in *Proc. 6th Int. Conf. Softw. Eng. Defence Appl.*, P. Ciancarini, M. Mazzara, A. Messina, A. Sillitti, and G. Succi, Eds. Cham, Switzerland: Springer, 2020, pp. 86–93, DOI: 10.1007/978-3-030-14687-0_8.

[5] K. Shu, S. Wang, D. Lee, and H. Liu, "Mining disinformation and fake news: Concepts, methods, and recent advancements, "in *Disinformation, Misinformation, and Fake News in social media. Cham*, Switzerland: Springer, pp.1–19, 2020, DOI: 10.48550/arXiv.2001.00623.

[6] N. Dehouche, "Plagiarism in the age of massive generative pre-trained transformers (GPT-3), "*Ethics Sci. Environ. Politics*, vol. 21, pp. 17–23, 2021, DOI: 10.3354/esep00195.

[7] R. Tang, and Y. N. Chuang, "The science of detecting llm-generated text." *Communications of the ACM* 6;, no. 4, pp. 50–59, 2024, DOI: 10.48550/arXiv.2303.07205.

[8] L. Floridi, and M. Chiriatti, "GPT-3: Its nature, scope, limits, and consequences." *Minds and Machines 30*; pp. 681–694, 2020, DOI: 10.1007/s11023-020-09548-1.

[9] J. Wu, L. S. Chao, S. Yang, Y. Yuan, and D. F. Wong, "A survey on LLM-generated text detection: Necessity, methods, and future directions." Computational Linguistics; pp. 1–65, 2025, DOI: 10.48550/arXiv.2310.14724.

[10] K. C. Fraser, H. Dawkins, and S. Kiritchenko, "Detecting ai-generated text: Factors influencing detectability with current methods." *Journal of Artificial Intelligence Research82*, pp. 2232–2278, 2025, DOI: 10.48550/arXiv.2406.15583.

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding, "*arXiv preprint; arXiv:1810.04805*, 2018, DOI: 10.48550/arXiv.1810.04805.

[12] Y. Liu, M. Joshi, D. Chen, M. Ott, N. Goyal, J. Du, O. Levy, and M. Lewis, L. Zettlemoyer, and V. Stoyanov. "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint; arXiv:1907.11692*, 2019, DOI: 10.48550/arXiv.1907.11692.

[13] J. Pennington, R. Socher, and C.D. Manning, "Glove: Global vectors for word representation," In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014, DOI: 10.3115/v1/D14-1162.

[14] H. Zhou, "Research of text classification based on TF-IDF and CNN-LSTM", In *journal of physics: conference series*; vol. 2171, no. 1, p. 012 021. IOP Publishing, 2022, **DOI**: 10.1088/1742-6596/2171/1/012021.

[15] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. A. Goldstein, "A watermark for large language models", In *International Conference on Machine Learning*. PMLR, 2023, **DOI**: 10.48550/arXiv.2301.10226.

[16] K. C. Fraser, H. Dawkins, and S. Kiritchenko, "Detecting ai-generated text: Factors influencing detectability with current methods", *Journal of Artificial Intelligence Research* 82: 2233–2278, 2025, **DOI**: 10.48550/arXiv.2406.15583.

[17] R. Zhang, S. Hussain, P. Neekhara, and F. Koushanfar, "Remark-llm: A robust and efficient watermarking framework for generative large language models", *arXiv preprint arXiv:2310.12362*, 2024, **DOI**: 10.48550/arXiv.2310.12362.

[18] X. Yang, W. Cheng, Y. Wu, L. Petzold, W. Y. Wang, and H. Chen, "Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text," *arXiv preprint arXiv:2305.17359*. 2023, **DOI**: 10.48550/arXiv.2305.17359.

[19] S. Gehrmann, and A. M. Rush, "Gltr: Statistical detection and visualization of generated text", *arXiv preprint; arXiv:1906.04043*, 2019, **DOI**: 10.48550/arXiv.1906.04043.

[20] Z. Guo, and S. Yu, "Authentigpt: Detecting machine-generated text via black-box language models denoising", *arXiv preprint arXiv:2311.07700.2023*, **DOI**: 10.48550/arXiv.2311.07700.

[21] P. Wang, L. Li, B. Jiang, D. Zhang, K. Ren, and X. Qiu, "SeqXGPT: Sentence-level AI-generated text detection", *arXiv preprint; arXiv: 2310.08903*, 2023, **DOI**: 10.48550/arXiv.2310.08903.

[22] S. Alam, and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis, "Computational and Mathematical Organization Theory, 2019, **DOI**: 10.1109/IISA62523.2024.10786699.

[23] C. Opar, "StyloAI: Distinguishing AI-generated content with stylometric analysis," In *International conference on artificial intelligence in education*, pp. 105–114. Cham: Springer Nature Switzerland, 2024, **DOI**: 10.48550/arXiv.2405.10129.

[24] N. T. Sivesind, "Chat GPT-Generated-Abstracts", Hugging Face, 2023.

[25] A. Dmonte, M. Zampieri, K. Lybarger, M. Albanese, and G. Coulter. "Classifying human-generated and ai-generated election claims in social media." *arXiv preprint arXiv:2404.16116,* 2024, **DOI**: 10.48550/arXiv.2404.16116.

[26] H. Touvron, L. Martin, K. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov *et al*. "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288* (2023), **DOI**: 10.48550/arXiv.2307.09288.

[27] Q. J. Albert, S. Alexandre, M. Arthur, B. Chris, and S. C. Devendra. "Mistral 7b", *arXiv preprint arXiv:2310.06825*, 2023., **DOI**: 10.48550/arXiv.2310.06825.

[28] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay. "The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only." *arXiv preprint arXiv:2306.01116*, 2023, **DOI**: 10.48550/arXiv.2306.01116.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, J. Vanderplas. "Scikit-learn: Machine learning in Python". The Journal of machine Learning research. 2011 Nov 1; 12:2825-30, **DOI**: 10.5555/1953048.2078195.

[30] S. Bird. "NLTK: the natural language toolkit". In *Proceedings of the COLING/ACL 2006 interactive presentation session*s 2006, pp. 69–72, **DOI**: 10.48550/arXiv.cs/0205028.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Corrado, S. Ghemawat. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". *arXiv preprint arXiv:1603.04467*. 2016, **DOI**: 10.48550/arXiv.1603.04467.

[32] M. Honnibal. "spaCy 2: Natural language understanding with Bloom embeddings", convolutional neural networks and incremental parsing. 2017.

[33] Textstat. https://textstat.org/.

[34] LanguageTool. https://pypi.org/project/language_tool_python/

[35] Pyspellchecker. https://pypi.org/project/pyspellchecker/

[36] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." *Journal of machine learning research 21*, no. 140, 1–67, 2020, **DOI**: 10.48550/arXiv.1910.10683.

Doaa Mostafa is a Ph.D. student in the computer science department at the faculty of computer and information science, Ain Shams University, Cairo, Egypt. She received her master's degree in computer science from the faculty of computer and information science, Ain Shams University in 2020. Her research interests include big data, artificial intelligence, natural language processing, and data mining.

Sally S. Ismail is a lecture of Computer Science at Ain Shams University, Cairo, Egypt. She holds a Ph.D. in Computer Science field from the University of Ain Shams. Her research interests include information retrieval, text summarization, and sentiment analysis.

Mostafa Aref is a professor of Computer Science at Ain Shams University, Cairo, Egypt. He holds a Ph.D. in engineering science in system Theory and engineering, June 1988, University of Toledo, Toledo, Ohio. He obtained his M.Sc. in computer science, 1983, University of Saskatchewan, Saskatoon, Sask. Canada. His research areas are natural language processing, knowledge representation, and ontology