Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064747

ARTICLE

Upholding Academic Integrity amidst Advanced Language Models: Evaluating BiLSTM Networks with GloVe Embeddings for Detecting AI-Generated Scientific Abstracts

Lilia-Eliana Popescu-Apreutesei, Mihai-Sorin Iosupescu, Sabina Cristiana Necula and Vasile-Daniel Păvăloaia*

Department of Accounting, Information Systems and Statistics, Faculty of Economics and Business Administration, Alexandru Ioan Cuza University of Iasi, Iasi, 700505, Romania

*Corresponding Author: Vasile-Daniel Păvăloaia. Email: danpav@uaic.ro

Received: 22 February 2025; Accepted: 12 May 2025; Published: 03 July 2025

ABSTRACT: The increasing fluency of advanced language models, such as GPT-3.5, GPT-4, and the recently introduced DeepSeek, challenges the ability to distinguish between human-authored and AI-generated academic writing. This situation is raising significant concerns regarding the integrity and authenticity of academic work. In light of the above, the current research evaluates the effectiveness of Bidirectional Long Short-Term Memory (BiLSTM) networks enhanced with pre-trained GloVe (Global Vectors for Word Representation) embeddings to detect AI-generated scientific abstracts drawn from the AI-GA (Artificial Intelligence Generated Abstracts) dataset. Two core BiLSTM variants were assessed: a single-layer approach and a dual-layer design, each tested under static or adaptive embeddings. The single-layer model achieved nearly 97% accuracy with trainable GloVe, occasionally surpassing the deeper model. Despite these gains, neither configuration fully matched the 98.7% benchmark set by an earlier LSTM-Word2Vec pipeline. Some runs were over-fitted when embeddings were fine-tuned, whereas static embeddings offered a slightly lower yet stable accuracy of around 96%. This lingering gap reinforces a key ethical and procedural concern: relying solely on automated tools, such as Turnitin's AI-detection features, to penalize individuals' risks and unjust outcomes. Misclassifications, whether legitimate work is misread as AI-generated or engineered text, evade detection, demonstrating that these classifiers should not stand as the sole arbiters of authenticity. A more comprehensive approach is warranted, one which weaves model outputs into a systematic process supported by expert judgment and institutional guidelines designed to protect originality.

KEYWORDS: AI-GA dataset; bidirectional LSTM; GloVe embeddings; AI-generated text detection; academic integrity; deep learning; overfitting; natural language processing

1 Introduction

The recent surge in advanced language model capabilities has introduced scenarios once confined to speculative fiction: machines now produce scientific abstracts, literature reviews, and entire research papers that appear convincingly human-like. OpenAI's GPT-3 and GPT-4 exemplify this generative power, offering coherent, contextually apt responses that rival or surpass human writing in surface quality [1,2] while newer entrants like DeepSeek and Gemini are rapidly gaining traction in the market, demonstrating advanced capabilities in both general language tasks [3] and specialized domains such as code intelligence [4]. Although the potential benefits for drafting, summarizing, and assisting in scholarly tasks are enticing, the

academic sphere faces a deep quandary—unregulated or undisclosed reliance on AI tools can easily erode foundational principles of originality and integrity.

This tension manifests most starkly in the assessment of student work and scholarly manuscripts. Traditional plagiarism-checking systems, which rely on cross-referencing known texts, struggle against AI's capacity to produce novel sequences that do not directly copy from any single source [5]. More recent attempts at AI-specific detection tools have shown promise but remain imperfect, with tendencies toward false alarms or missed instances [6]. Against this backdrop, researchers are compelled to explore robust computational strategies that can better delineate human-authored text from machine-generated output.

The primary objectives of this research are as follows:

1. Develop a Deep Learning (DL) model by creating a BiLSTM-based model utilizing GloVe embeddings that can effectively distinguish between human-authored and AI-generated scientific abstracts.
2. Evaluate embedding configurations to compare the performance of models with trainable embeddings vs. those with non-trainable embeddings, examining how each approach affects accuracy and the risk of overfitting.
3. Highlight the inability to achieve 100% accuracy by demonstrating that, despite advanced modeling techniques, achieving perfect accuracy in detecting AI-generated text remains elusive, underscoring the need for caution in relying solely on automated tools for punitive measures.
4. Advocate for comprehensive evaluation practices, to recommend that academic institutions combine technological solutions with human oversight and policy development to ensure fair and accurate assessments of scholarly work.

1.1 State-of-the-Art in AI-Generated Text Detection in Student Papers

The scholarly discourse on AI-generated content has intensified as language models have grown more sophisticated. The ability of GPT-class models to produce coherent, contextually apt text has alarmed educators and researchers, who worry about the erosion of authentic intellectual engagement. Cotton et al. [7] documented how these tools create uncertainty in academic evaluations, blurring lines between original and AI-assisted contributions. Bhullar et al. [8] identify this predicament in higher education, where students might exploit AI text generators for assignments, undermining critical thinking and personal mastery of material.

Limitations in detection tools exacerbate the situation. Standard plagiarism detectors, tuned to spot copied or paraphrased content, prove inadequate when faced with genuinely novel, AI-fabricated prose [5]. Attempts to refine AI detection have shown some promise. For instance, Theocharopoulos et al. [9] reported 98.7% accuracy using LSTM networks and Word2Vec embeddings. Yet even these advancements do not close the gap entirely, especially considering constraints such as computational resources and the ever-evolving ingenuity of language models.

Although a scarcity of studies was published before 2024 that specifically address AI-generated text detection in student papers, several recent works have investigated the broader challenge of detecting AI-generated texts across various domains, and these address concepts closely related to the study. Thus, in the last years, there has been research that highlighted the growing challenge of distinguishing AI-generated texts from human-written content, emphasizing both technological and evaluative limitations. Hakam et al. [10] found that neither researchers nor AI-detection software could reliably identify LLM-generated abstracts, underscoring risks of false positives/negatives and the need for advanced detection methods like BiLSTM networks enhanced with GloVe embeddings. Similarly, Lawrence et al. [11] reported that while ChatGPT-generated abstracts were rated as lower in quality, evaluators' confidence in their authorship remained

comparable to human-written texts, revealing the subjectivity and inconsistency of manual assessments. Further complicating detection, Nabata et al. [12] observed that only 40% of participants correctly identified human abstracts, with 63% preferring AI-generated versions, suggesting inherent biases in human evaluation. Kim et al. [13] reinforced these concerns, noting modest AI-detection rates (56%–87%) and low human accuracy (53.8%), which collectively underscore the inadequacy of current tools. Shcherbiak et al. [14] added that while human reviewers and GPTZero outperformed GPT-4 in detection tasks, inter-rater agreement remained poor, highlighting systemic reliability issues. Finally, Cheng et al. [15] demonstrated that although human evaluators achieved 93% detection accuracy, AI-generated abstracts exhibited significantly lower quality, implying that qualitative analysis could complement automated methods. Together, these findings illustrate the urgent need for robust, hybrid frameworks that integrate advanced computational models with human oversight to mitigate risks to academic integrity.

Based on the importance of this topic for academics and in the pursuit of elaborating on the superiority of DL techniques vs. the traditional ones, the recent advancements in detecting AI-generated academic text are vividly illustrated by recently released studies. Chowdhury et al. [16] report outstanding performance of DL approaches, noting that the majority of participating systems utilized fine-tuned transformer-based models, with top-performing systems achieving F1 scores exceeding 0.98. Furthermore, it explicitly states that "Nearly all submitted systems outperformed the n-gram-based baseline," supporting the superiority of these modern techniques over traditional n-gram methods. Specific system descriptions within the next research showcase the sophistication and effectiveness of current DL strategies. Jiao et al. [17] achieved a near-perfect F1 score (0.999) by leveraging features extracted from a Large Language Model (LLM), namely Llama-3.1-8B, as a proxy, without the need for fine-tuning, and classifying these features with a Convolutional Neural Network (CNN). Other highly successful approaches involved fine-tuning "cutting-edge transformer-based models," reaching F1 scores around 0.96–0.97 [18]. Additionally, fusion models combine pre-trained language model embeddings with carefully engineered stylometric and linguistic features. They address known limitations of previous detectors, such as high false-positive rates [19]. DL models dominated the top rankings, even highly optimized feature-based methods achieved competitive F1 scores (e.g., 0.986 reported by [20]), demonstrating the high performance benchmark set by current methodologies in education. Collectively, these recent findings demonstrate the outstanding performance and rapid evolution of DL strategies for AI-generated text detection, significantly advancing the state-of-the-art compared to traditional techniques.

Some other studies included in their analysis prove that state-of-the-art performance in AI-generated text detection is often achieved by DL algorithms that integrate semantic understanding with stylistic feature analysis. Additionally, other manuscripts highlight inherent stylistic differences exploitable by detection systems. For example, Varadarajan et al. [21] found that AI-generated text exhibits remarkably limited variance in inferred psychological traits compared to human writing, offering a potential avenue for unsupervised, style-based detection. However, the robustness of these detectors is challenged by the adversarial evasion techniques, where Creo [22] demonstrated that homoglyph-based attacks can systematically circumvent seven state-of-the-art detectors (including watermarking and transformer-based tools), reducing their Matthews Correlation Coefficient from 0.64 to −0.01. This vulnerability arises because homoglyphs disrupt tokenization and feature extraction pipelines, undermining both semantic and stylistic analysis. Although highly tuned transformer models alone can achieve strong results according to [18], the trend towards incorporating stylistic analysis alongside deep semantic representations underscores the sophistication of current methods. The above-mentioned advanced DL strategies, leveraging the use of homoglyph as well as semantic depth and stylistic nuances, demonstrate great performance that surpasses traditional techniques, addressing the ongoing challenge and need for AI text detection in Academia.

The increasing sophistication of AI-generated text has highlighted the urgent need for enhanced automated detection tools, as current models still struggle with false positives and domain-specific adaptation. Kim et al. (2024) [13] found that while ChatGPT-generated abstracts were well-formatted, they exhibited high plagiarism rates (20%–32%) and AI-detection scores (63%–87%), necessitating specialized models like BiLSTM with GloVe to minimize misclassifications.

1.2 Ethical Considerations and Academic Misconduct

However, beyond technical advancements, ethical considerations underscore the need for a hybrid evaluation approach. Khlaif et al. [23] stressed that while AI-generated texts maintain high quality, they introduce significant integrity concerns, including plagiarism and authorship ambiguity, making transparent policies and human oversight essential. Shcherbiak et al. [14] supported this view, suggesting that AI can serve as a prescreening tool but should not replace expert judgment to prevent systematic errors. Similarly, Cheng et al. [15] emphasized that the often low quality of AI-generated abstracts reinforces the necessity of manual verification, even when automated detectors perform well. The study developed by Kumar et al. [24] highlights the need for ethically grounded strategies in leveraging AI/GPT technology for education, emphasizing that while AI can enhance learning, the role of human educators remains crucial and should not be overshadowed by the potential benefits of technology.

The emergence of AI language models continues to raise significant concerns within academic institutions, primarily due to their capacity to generate original content that students can easily plagiarize with a reduced risk of detection [25]. This capability challenges traditional understandings of academic misconduct. While previous research highlighted the role of perceived costs and benefits in students' decisions to engage in academic misconduct, the diminished detectability of AI language models-based plagiarism lessens the perceived cost of being caught, thereby weakening the explanatory power of rational choice theory in this context [25]. The above-mentioned authors argue that moral disengagement is a key factor influencing AI language models-based plagiarism, with perceived benefits, punishment severity, and informal sanctions also playing significant roles.

Furthermore, moral disengagement can amplify the impact of formal sanctions on AI language models-based plagiarism. Wang and Cornely [26] highlight that ChatGPT's advanced AI capabilities, while offering transformative potential, have led to an increase in academic misconduct. Students are exploiting this technology to complete assignments, fabricate essays, and even cheat during examinations, which undermines the fundamental principles of educational integrity. This necessitates a comprehensive strategy by academic institutions to address these new forms of academic dishonesty, balancing technological advancements with the need to uphold academic integrity [26].

These trends align with ethical concerns raised by Májovsky et al. [27], who demonstrated that ChatGPT can generate fraudulent medical articles indistinguishable from human work, risking academic integrity. Ethical imperatives, such as mitigating hallucination risks in AI-generated references [28], further reinforce the need for a balanced strategy to uphold academic authenticity.

1.3 The Recent Outstanding Advances of DL for AI-Generated Text Detection

The recently published research reflects a sustained effort toward developing and evaluating sophisticated techniques for detecting AI-generated text, with a notable emphasis on DL algorithms that have been made, due to their potential to outperform traditional methods. The need for ongoing reassessment towards detection capabilities in light of evolving AI models is a recurring theme, highlighted by researchers like Wang and Zhou [29]. Other studies focus on the evaluation and comparison of different methodologies, and in this respect, Onan and Çelikten [30] explored the effectiveness of various text representations coupled

with both DL and Machine Learning (ML) classifiers specifically for identifying AI-generated scientific abstracts. Extending the scope, Nabata et al. [12] evaluated several detection approaches within the context of software reviews, indicating the broad applicability and ongoing refinement of these techniques across different domains.

The following analyzed studies, primarily published in early 2025, strongly affirm the central importance and outstanding performance of transformer-based DL architectures in the detection of AI-generated text. Although not all of the studies reviewed focus explicitly on the detection of AI-generated text within student essays—the primary applied context of this research—the following analysis aims to highlight ongoing efforts by researchers to leverage DL methodologies in pursuit of high detection accuracy.

These models, particularly when fine-tuned, consistently establish high-performance benchmarks. For example, Maktabdar Oghaz et al. [31] reported "exceptional performance," achieving near-perfect F1 scores (around 0.99) using custom RoBERTa and DistilBERT models for classifying ChatGPT-generated content, setting a significantly high baseline for current detection capabilities. Further evidence for the general effectiveness of fine-tuning standard transformers like BERT, DistilBERT, and RoBERTa comes from Yadagiri et al. [32], who demonstrated their success with optimized hyperparameters for the general text detection challenge.

Based on the results published in recent manuscripts, RoBERTa proves to be a prominent transformer variant, as it has been investigated extensively, showcasing its effectiveness through various approaches. Beyond using custom models [31] and standard fine-tuning [32], more sophisticated strategies involving RoBERTa have proven effective. Mobin and Islam [33] demonstrated the utility of RoBERTa-based ensembles combined with specialized weighting techniques (such as inverse perplexity) to enhance robustness and generalization across diverse text domains, achieving strong results even against adversarial manipulations in competitive evaluations. Moreover, the critical importance of optimizing the fine-tuning process for established encoder models like RoBERTa (and XLM-R) was underscored by Agrahari et al. [19]. Their experiments revealed that careful selection of training epochs, maximum input size, and techniques for handling class imbalance could improve performance by a significant 5–6% in absolute terms, highlighting the crucial role of hyperparameter tuning in building effective and scalable RoBERTa-based detection systems. For the case of multilingual settings, Marchitan et al. [34] employed Low-Rank Adaptation (LoRA) to fine-tune XLM-Roberta-Base, achieving competitive results while maintaining computational efficiency.

DistilBERT is also highlighted as another widely used transformer, and demonstrates spectacular versatility and effectiveness across various detection scenarios. Thus, DistilBERT (similar to RoBERTa) achieved high performance in custom configurations for classifying ChatGPT content [31] and showed effectiveness in general text detection via standard fine-tuning [32]. Specialized applications include DistilBERT successful use by Abiola et al. [35] to differentiate between human-written and machine-generated text written in English, achieving notable performance on this binary classification task. Yadagiri et al. [32] applied DistilBERT (alongside XLM-RoBERTa) to achieve high accuracies (up to 92.2%) in identifying AI-generated essays. Moreover, Yadagiri et al. [32] employed the DistilBERT-based framework to tackle the challenge of cross-domain machine-generated text detection, focusing on robustness against adversarial manipulations, further highlighting the capabilities of this architecture.

Research extends beyond single-model fine-tuning, exploring diverse architectures, hybrid methods, and innovative techniques. Other transformer variants like XLM-RoBERTa were utilized by Yadagiri et al. [32] for AI essay detection and highlighted by Agrahari et al. [19] regarding the importance of optimized fine-tuning parameters. Marchitan et al. [34] achieved top results by exploring both masked language models and large causal models like Qwen. Their approach involved efficiently fine-tuning only the last layer and classification head of the Qwen2.5-0.5B model, demonstrating the potential of partial model adaptation for

detection tasks. Hybrid models combining multiple architectures have shown promise. Mohamed et al. [36] demonstrated that a hybrid model integrating RoBERTa, T5, and GPT-Neo achieved an impressive 99% detection accuracy, although they noted that further adaptation for specialized fields such as medicine remains crucial. Consequently, for academic essay authenticity detection, AL-Smadi [37] achieved high results (F1 scores of 99.7% and 98.4%) by combining fine-tuned ELECTRA/AraELECTRA transformer models specifically with stylometric features. Such hybrid approaches, mentioned above, significantly outperformed simpler baselines.

Integrating diverse feature types is another avenue for improvement. Zhang et al. [38] showcased the benefit of combining semantic features derived from RoBERTa with probabilistic (stylistic) features extracted from LLaMA3, leading to strong classification performance. Innovative training strategies are also being developed. Emi et al. [39] introduced an active learning approach specifically for cross-domain machine-generated text detection. Their method, leveraging a two-stage training procedure involving mining high-error examples and retraining the model to mitigate undertraining, achieved state-of-the-art performance on adversarial attack benchmarks, showcasing the potential of iterative learning strategies. Across these varied approaches, the significance of meticulous hyperparameter tuning, as emphasized by Singh et al. [19], remains a critical factor for optimizing DL-based detection systems.

These findings collectively highlight the power and adaptability of modern DL techniques in the crucial challenge of AI-generated text detection, showcasing advancements in model architectures, training strategies, and domain adaptation.

1.4 Mapping the Current within Existing Research

Further analysis of recent studies spotlights the growing challenges and advancements in detecting AI-generated scientific abstracts, aligning closely with the focus of this research on BiLSTM networks with GloVe embeddings. Mese [40] identified a significant rise in AI-generated content probabilities (3.8% to 5.7%) between 2022 and 2023 using a detection tool with 97.06% accuracy, emphasizing the need for reliable methods like the proposed BiLSTM-GloVe model. Human evaluators' limitations are starkly evident as Makiev et al. [41] reported that orthopaedic experts identified AI-generated abstracts with only 31.7%–34.9% accuracy, while AI detectors achieved 42.9%–66.6% accuracy, highlighting the critical gap that advanced models must address. Similarly, Alencar-Palha et al. [42] found low human sensitivity (58%) and specificity (62%) in distinguishing AI-generated dental abstracts, further justifying the shift toward automated solutions.

The urgency for robust detection tools is amplified by trends showing increased AI adoption. In light of the above, Carnino et al. [43] observed a post-ChatGPT surge in AI-generated text in otolaryngology abstracts (34.36% to 46.53%), while Howard et al. [44] noted a 2.37-fold increase in AI content in ASCO abstracts by 2023. Advanced detection frameworks are emerging to counter these challenges. Gralha et al. [45] developed a classifier with 97%–99% accuracy using decision trees and Scikit-learn, while Hamed and Wu [46] introduced xFakeSci, achieving F1 scores of 80%–94% through network models and calibration heuristics. These studies validate the feasibility of high-accuracy models like the BiLSTM-GloVe approach, which achieved 96%–97% accuracy but emphasized the need to balance overfitting risks (trainable embeddings) and generalization (non-trainable embeddings). Weber-Wulff et al. [6] found that numerous AI-detection tools struggle to surpass 80% accuracy, often biased toward misclassifying authentic text as AI-generated. Gao et al. [47] documented the inconsistency of tools in detecting synthetic scientific abstracts, particularly when AI content was paraphrased or translated. Thus, the consensus emerges: purely automated tools cannot unerringly separate human from machine authorship. A more nuanced, multifaceted approach that marries technology with informed human judgment seems inevitable.

Finally, hybrid evaluation frameworks—integrating automated detection with human oversight—are increasingly advocated. Makiev et al. [41] and Alencar-Palha et al. [42] both highlighted human evaluators' biases and inconsistencies, supporting this study's recommendation to combine BiLSTM outputs with transparent policies and expert judgment.

The existing body of recent literature accentuates the complexities and ethical dilemmas posed by AI-generated content in academia, emphasizing the need for robust detection mechanisms and balanced evaluative frameworks. Therefore, this research proceeds to investigate the efficacy of BiLSTM networks with GloVe embeddings in detecting AI-generated scientific abstracts, aiming to provide a valuable contribution towards upholding academic integrity in this evolving landscape.

2 Methodology

2.1 Data Collection and Preprocessing

The research employed the AI-GA dataset [48] comprising 28,662 scientific abstracts equally divided between human-authored and GPT-3-generated texts. The human-written abstracts, drawn from COVID-19 research, ensure topical relevance and consistency. For each human-authored abstract, GPT-3 generated a corresponding AI abstract using the same title, thus enabling direct, controlled comparisons.

Data preprocessing (Fig. 1) directly affects the model's ability to identify subtle distinctions between human and AI-generated text. It removed extraneous HTML tags, special characters, numbers, and excessive whitespace to yield cleaner input. Text was normalized to lowercase, and stopwords were removed [49] to reduce noise and focus the model on meaningful content. Subsequently, each abstract was tokenized into discrete lexical units, with sequences standardized through padding or truncation to accommodate batch processing requirements. Quantitative analysis of the corpus revealed abstracts averaging 959.0 characters for human-authored texts and 891.0 characters for AI-generated counterparts, translating to approximately 168 and 156 tokens per abstract, respectively, when using standard English tokenization assumptions. Based on the above mentioned measurements, the authors established the maximum sequence length of 200 tokens, providing sufficient accommodation for most abstracts while minimizing unnecessary padding. Only 17.3% of the sequences required truncation at this threshold.

For word representation, authors implemented and compared two distinct embedding methodologies:

1. GloVe Embeddings [50]—the Stanford NLP Group's Global Vectors for Word Representation uses 100-dimensional vectors trained on a corpus of 6 billion tokens from Wikipedia and Gigaword 5. GloVe embeddings operate on the principle of factorizing a word-context co-occurrence matrix, capturing both global statistical information and local contextual relationships.
2. Word2Vec Embeddings [51]—300-dimensional vectors derived from the Google News Corpus, which use Neural Network architectures to learn word associations from large text corpora, emphasizing local context windows through either continuous bag-of-words or skip-gram approaches.

Both embedding strategies were evaluated under two configurations:

1. Static Embeddings (trainable = False): In this configuration, the pre-trained vectors remained unchanged during model training. This approach represents pure transfer learning, leveraging the established semantic relationships without modification. Its advantages include reduced computational demands, prevention of catastrophic forgetting of general language knowledge, and mitigation of overfitting risks by constraining the parameter space.
2. Dynamic embeddings (trainable = True): Implementation involved leveraging pre-trained word embeddings as initial weights within the embedding layer. This constituted an adaptive transfer learning strategy, enabling the model to fine-tune the embedding space and capture nuanced, domain-specific

semantic relationships prevalent in the scientific abstract corpus. While this specialization held the potential to enhance task performance, it inherently increased the model's parameter space, consequently elevating the susceptibility to overfitting. This fine-tuning mechanism was selectively applied to GloVe embeddings, informed by preliminary experimental results that demonstrated negligible performance gains from training Word2Vec embeddings, notwithstanding their significantly greater computational cost.

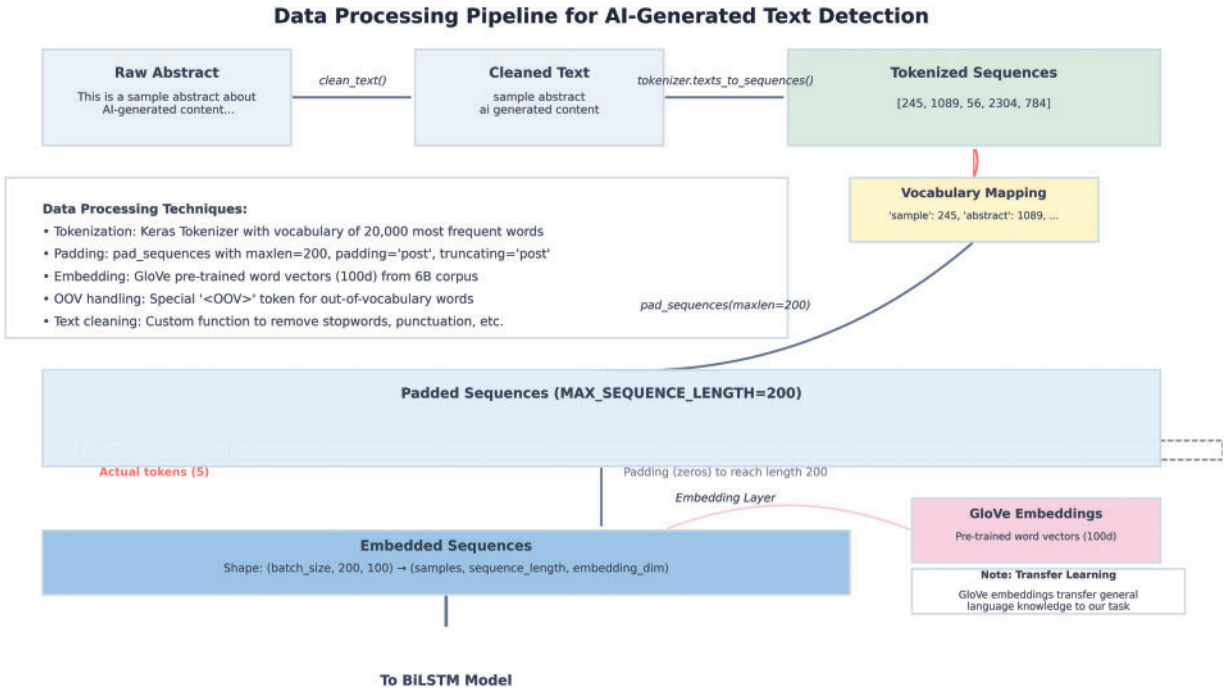

Figure 1: Tokenization and embedding pipeline for AI-generated text detection

The vocabulary was constructed by mapping each token to a numerical index and was limited to 20,000 tokens—a threshold established by analyzing the frequency distributions in the corpus to balance coverage against computational efficiency and diminishing returns associated with extremely rare terms.

2.2 Model Architecture

The proposed architecture incorporates a BiLSTM network to facilitate the contextual analysis of sequential textual data, enabling the model to leverage information from both preceding and subsequent elements within each sequence [52]. This design acknowledges that natural language often contains cues that appear either before or after a given token, making a unidirectional pass insufficient in certain instances. At the heart of this arrangement is the LSTM cell, introduced by Hochreiter and Schmidhuber [53], which addresses the shortcomings of vanilla Recurrent Neural Networks (RNNs) in retaining long-range dependencies.

An LSTM cell employs a gated architecture—comprising a forget gate, input gate, and output gate—to regulate the flow of information through the network. The forget gate discards irrelevant past states while the input gate determines which new information to store and the output gate controls how much of the hidden state is passed forward. By allowing these selective updates, LSTM cells mitigate the vanishing and exploding gradient issues frequently encountered in conventional RNNs. When arranged bidirectionally (i.e.,

processing the sequence in both forward and backward directions), the network can extract richer contextual signals surrounding each token, thereby enhancing its representational capacity [52].

The proposed BiLSTM architecture (check Fig. 2) begins with an Embedding layer initialized using pre-trained GloVe embeddings [50], which serves as a lookup table mapping each token to its corresponding vector representation. Depending on whether the embedding is set to "trainable = True" or "trainable = False," the network may further refine or preserve these word vectors during training. Immediately following the embedding, a SpatialDropout1D layer prevents overfitting by randomly zeroing entire feature maps, encouraging reliance on distributed activations rather than overly specific patterns. Next, the first BiLSTM layer (256 units) is configured with return_sequences = True to forward a series of hidden states to the subsequent recurrent block, thereby maintaining sequential continuity. The dropout rate within the LSTM cell (0.1) and the recurrent dropout rate (0.5) further prevent overfitting by deactivating selected neural connections in both the input and recurrent pathways. Subsequently, Batch Normalization [54] is applied to stabilize shifting input distributions. The second BiLSTM layer, containing 32 units, refines the higher-level representations by discarding unhelpful features while reinforcing salient sequence patterns. A subsequent batch normalization layer harmonizes the activation magnitudes, aiding smoother convergence. The model then transitions into a Dense layer with 128 units, compacting the learned temporal features into a form suitable for final classification. Applying dropout (0.3 rate) both before this dense transformation and before the final output provides an additional safeguard against overfitting. Finally, a single sigmoid neuron produces a probability score for binary classification.

2.3 Hyperparameter Search and Structural Evaluation

Neural Network performance depends heavily on architecture configuration and hyperparameter selection—especially in sequence models like BiLSTMs, where multiple design choices interact to create a vast combinatorial search space. Our dual approach consisted of a systematic exploration via Hyperband optimization followed by targeted ablation studies to both identify optimal configurations and understand the contributions of architectural components.

2.3.1 Hyperband Optimization Framework

Traditional grid search or manual tuning methods are computationally prohibitive for DLmodels with numerous hyperparameters. Given the inherent computational limitations imposed by the Kaggle environment—namely, constrained GPU memory and limited session durations—it was imperative to adopt an optimization strategy that maximized exploration while judiciously allocating resources. The Hyperband algorithm [55] synthesizes the concept of successive halving with ideas from multi-armed bandit formulations. In essence, it adaptively distributes computational resources (e.g., the number of training epochs) across different hyperparameter configurations. Models exhibiting strong preliminary performance receive additional resources, whereas those with weaker metrics are curtailed early. By prioritizing promising configurations, Hyperband can explore a much broader set of hyperparameter values than would be feasible under a fixed computational budget using traditional search strategies.

For building BiLSTM architecture, the authors conducted 16 trials, examining multiple hyperparameter dimensions (see Table 1).

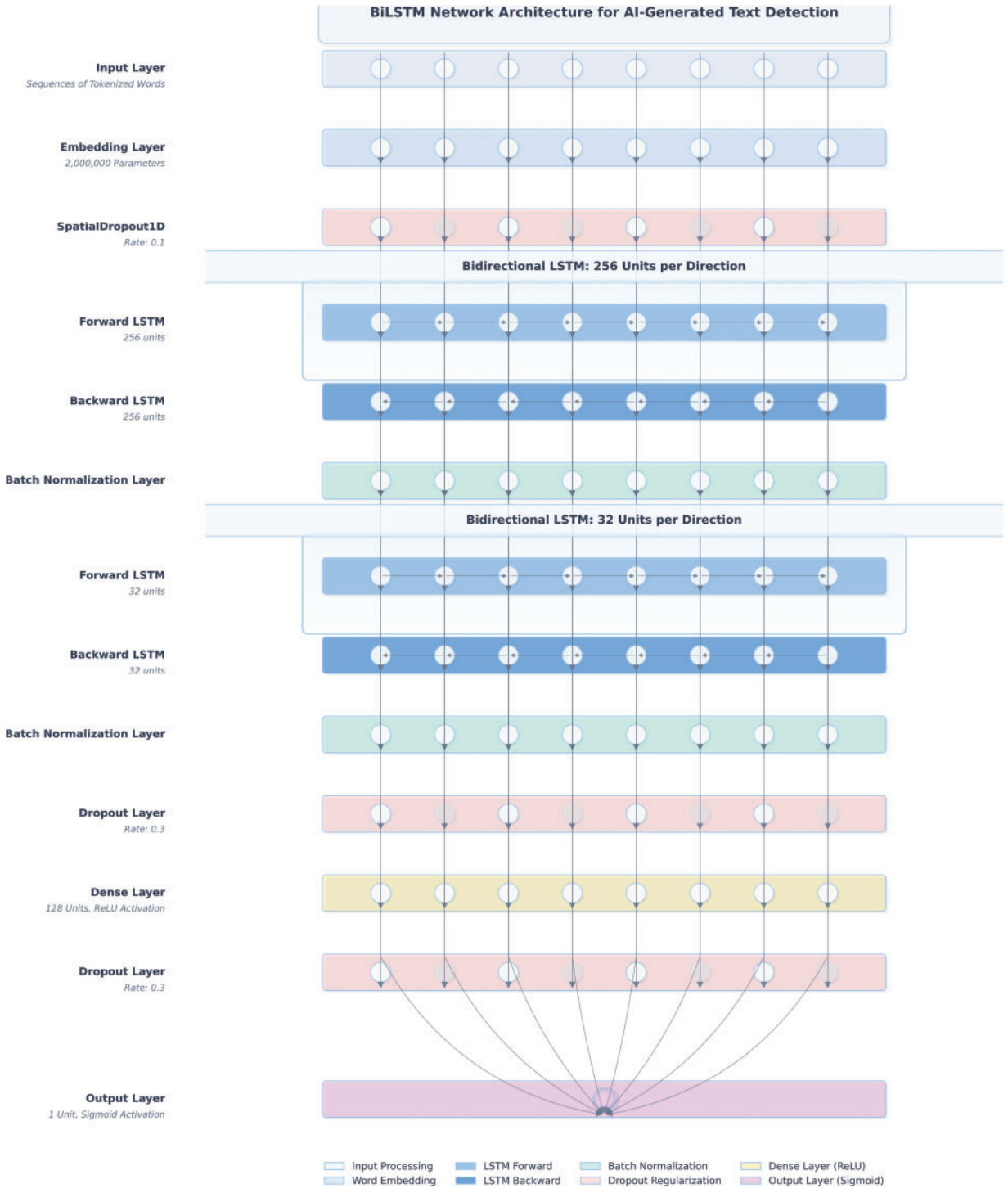

Figure 2: Dual BiLSTM model architecture with GloVe embeddings for binary text classification

Table 1: Hyperparameter configuration space for BiLSTM model optimization

Parameter category	Parameter name	Search domain	Description
Network Structure	First Layer LSTM Units	{64, 128, 192, 256}	Number of units in the first bidirectional LSTM layer.
	Second Layer LSTM Units	{32, 64, 96, 128}	Number of units in the second bidirectional LSTM layer.
	Dense Layer Units	{32, 64, 96, 128}	Number of units in the fully connected layer preceding the output.
Regularization	Spatial Dropout Rate	{0.1, 0.2, 0.3, 0.4, 0.5}	Dropout rate applied to entire feature maps in the spatial dropout layer.
	LSTM Dropout	{0.1, 0.2, 0.3, 0.4, 0.5}	Dropout rate applied to input connections of LSTM layers.
	LSTM Recurrent Dropout	{0.1, 0.2, 0.3, 0.4, 0.5}	Dropout rate applied to recurrent connections within LSTM cells.
	Dense Dropout	{0.3, 0.4, 0.5, 0.6, 0.7}	Dropout rate applied after the dense layer.
	Output Dropout	{0.3, 0.4, 0.5, 0.6, 0.7}	Dropout rate applied before the output neuron(s).
Optimization	Learning Rate	10^{-2}, 10^{-3}, 10^{-4}	Step size used during gradient descent optimization.

Each configuration underwent early stopping (patience = 3 on validation loss) to prevent overfitting while maximizing computational efficiency.

2.3.2 Component-Wise Ablation Methodology

Beyond hyperparameter tuning, it is often necessary to dissect model components to understand their individual contributions to overall performance—a process known as an ablation study [56]. Ablation studies involve selectively modifying or removing specific architectural elements (e.g., spatial dropout layers, batch normalization layers, or entire recurrent layers) and quantifying how these changes affect performance metrics. While hyperparameter optimization focuses on fine-tuning numerical values, ablation studies address structural decisions, revealing which segments of the architecture offer substantial contributions.

The proposed ablation study focused on the following two fronts:

1. Hyperparameter Sensitivity Analysis: A targeted investigation into the model's sensitivity to learning rates $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and batch sizes {32, 64, 128, 256}. For each combination, authors trained

a BiLSTM model and evaluated performance on a held-out validation set using accuracy and F1-score metrics.

2. Architectural Component Analysis: A separate process involving strategic removals or modifications of architectural elements (e.g., dropout layers, normalization strategies, and the number of LSTM layers).

Following this two-pronged ablation study, researchers concretized the various architectural configurations under scrutiny. In particular, they examined six structural variants—from a simple baseline LSTM to a fully optimized BiLSTM approach—under both static and dynamic embedding settings (Table 2).

Table 2: BiLSTM architectural variants

Model variant	Embedding trainable?	Spatial dropout	LSTM layers	Normalization	Notable differences
Baseline	True/False	None	1× Unidirectional LSTM (256 units)	None	Represents a simple foundational model. Comprises a single LSTM layer (256 units), followed by a dense layer (128 units), concluding with a sigmoid output.
No Spatial Dropout	True/False	None (Explicitly)	2× Bidirectional LSTM (256 units & 32 units)	None	Eliminates the SpatialDropout1D layer found in certain variants. Instead, stacks two BiLSTM layers; first with 256 units, then a second with 32 units.
No Batch Norm	True/False	Present (0.1 rate)	2× Bidirectional LSTM (256 units & 32 units)	Omitted	Preserved spatial dropout while removing batch normalization. Primarily aims to gauge how batch normalization affects stability and performance.
Single LSTM	True/False	Present (0.1 rate)	1× Bidirectional LSTM (256 units)	Batch Normalization (1 layer)	Consolidates the architecture to a single BiLSTM layer rather than two. Useful for understanding if a deeper LSTM stack substantially improves classification.
Full Model	True/False	Present (0.1 rate)	1× BiLSTM (256 units, return sequences), followed by 1× BiLSTM (32 units) with Dropout & Recurrent Dropout as tuned	Batch Normalization (2 layers)	Incorporates all hyperparameters from the Hyperband outcomes, including double BiLSTM, batch normalization, and dropout. Represents the most feature-rich.
Optimized Model	True/False	None (Removed)	1× BiLSTM (256 units, return sequences), followed by 1× BiLSTM (32 units) with Dropout & Recurrent Dropout as tuned	Batch Normalization (2 layers)	Similar to the Full Model but excludes spatial dropout, reflecting ablation findings that suggested better performance without it.

In each variant, the dense layers following the LSTM blocks typically involve a 128-unit dense layer, sometimes accompanied by dropout before the final sigmoid output. For approaches retaining spatial dropout, the dropout rate was selected based on Hyperband tuning. Models labeled "trainable = True" or "trainable = False" indicate whether the embedding layer's weights are updated during backpropagation, allowing a direct comparison between frozen (pre-trained only) and learnable embeddings.

2.4 Model Training and Performance Evaluation

After refining the architectural and hyperparameter configurations via Hyperband optimization and ablation studies, the next phase involved systematically training each candidate model and assessing

generalization performance on unseen data. To accomplish this, the authors employed the binary cross-entropy objective [57]—appropriate for binary classification tasks—with model weights updated via the Adam optimizer [58] at a base learning rate of 0.001, as established by prior experiments.

All models benefited from a suite of callbacks designed to prevent overfitting and guide training:

1. Early Stopping [59]: Monitors validation loss and halts training once incremental improvement stagnates (patience parameter).
2. Model Checkpointing: Saves the model parameters at the epoch yielding the minimal validation loss, thereby preventing any subsequent regression in performance from overwriting an optimal checkpoint.
3. Learning Rate Reduction on Plateau [60]: Monitors validation loss and reduces the learning rate by a factor whenever improvement stalls for a specified number of epochs (patience), allowing the optimizer to fine-tune the model weights.
4. Learning Rate Scheduler: In some configurations, a custom scheduler exponentially decreased the learning rate past a certain epoch, fine-tuning updates as training proceeded.
5. Resource Logging: For deeper insight, a custom callback recorded GPU and CPU resource usage, which proved beneficial when operating under limited Kaggle GPU budgets.

The trained models were evaluated on a held-out test set comprising 20% of the data (with an additional 10% of the training split serving as a validation set). To comprehensively assess model performance, it was employed a range of evaluation metrics:

- Accuracy: The overall proportion of correct predictions made by the model.
- Precision: The ratio of true positive predictions to the total number of positive predictions made by the model, indicating the model's ability to avoid false positives.
- Recall (Sensitivity): The ratio of true positive predictions to the total number of actual positive instances, reflecting the model's ability to detect all positive instances.
- F1-Score: The harmonic means of precision and recall, providing a balance between the two metrics.
- Confusion Matrix Analysis: A detailed breakdown of true positives, true negatives, false positives, and false negatives, offering insights into the types of errors made by the model.

For more stringent evaluation, certain setups underwent 5-fold Stratified Cross-Validation, ensuring balanced class distribution in each fold. In this procedure, one fold is used as the test set while the remaining folds serve as training/validation data, and the final metrics are averaged across all folds. Additionally, extended training (up to 20 epochs) was explored for selecting GloVe-based BiLSTM variants.

2.5 Benchmarking against Baseline Approaches

In addition to the BiLSTM-based architectures described above, our methodology also involved constructing and evaluating a collection of baseline approaches ranging from traditional ML classifiers to simplified DL modules. This comparative exploration positions the proposed models in a broader context and illustrates how classic algorithms or less complex neural networks perform relative to the hyperparameter-optimized, ablated BiLSTMs.

For traditional ML classifiers, authors of the current study combined feature extraction methods (e.g., TF-IDF or simple Bag-of-Words) with widely recognized algorithms:

1. Logistic Regression [61]: A linear model that optimizes a logit-based cost function, offering interpretable decision boundaries and being adept at classification tasks with high-dimensional sparse features.
2. Naïve Bayes [62]: A probabilistic model under the simplifying assumption that input features are conditionally independent, typically efficient and surprisingly effective in text classification.

3. Random Forest [63]: An ensemble of decision trees that harnesses data and feature subsampling, thereby reducing variance. While random forests can handle a variety of data distributions, they can become quite large in parameter count.
4. Gradient Boosting Machines [64]: An iterative ensemble strategy that adds weak learners (usually small decision trees) stage by stage, each attempting to correct errors left by the preceding trees.
5. XGBoost [65]: A highly optimized gradient boosting library that can substantially outperform less specialized implementations of boosted decision trees, especially when carefully tuned.

Textual input for these pipelines was first transformed using either TF-IDF or Count Vectorizer methods, restricting feature numbers for computational tractability. Each model was then fit to the extracted feature vectors, with hyperparameters (e.g., the number of estimators or maximum iterations) modestly optimized to ensure stable convergence within environmental constraints. For every baseline, test predictions were recorded and evaluated using the same performance metrics (accuracy, F1, precision, recall).

On the simplified DL front, researchers introduced two smaller architectures:

1. CNN-based Model: A one-dimensional convolutional layer with a fixed kernel size, followed by a global max-pooling mechanism [66]. This configuration retains the ability to detect n-gram-like patterns in text while avoiding the parameter scale of recurrent networks.
2. Unidirectional LSTM [67]: A single-layer LSTM that processes input sequences from left to right, thus removing the bidirectional capacity. Though more compact than a BiLSTM, it provided an instructive baseline regarding how much bidirectionality improves classification of textual abstracts.

All baseline models were evaluated using the same train–test partitioning to facilitate a consistent and transparent comparison.

2.6 Statistical Evaluation and Significance Testing

Beyond conventional metrics, a robust experimental design calls for statistical checks that determine whether observed differences between model predictions are likely to persist beyond the particularities of a single sample. In this part, authors incorporated the several statistical tests and validation strategies, as follows:

1. McNemar's Test [68]: Specifically tailored for paired nominal data, McNemar's test scrutinizes two different classifiers' predictions on the same test set. A 2×2 contingency table counts how often each model is correct or incorrect on the same instance. If one model repeatedly misclassifies cases that the other handles correctly, McNemar's statistic flags whether the improvement is statistically nonrandom.
2. Paired t-tests for K-fold Cross-Validation [69,70]: When K-fold cross-validation was employed, it captured performance metrics (accuracy, F1, etc.) across each of the K folds. The distribution of these metrics allowed for paired t-tests that compare two models' fold-wise outcomes. Alignment of each fold's training and validation sets, was performed to minimize confounding factors arising from partition variability.
3. Confidence Intervals for Cross-Validation Metrics: To quantify uncertainty in mean performance, 95% confidence intervals were calculated around the cross-validation metrics. This measure is beneficial because a mere average can be misleading. Intervals illustrate the plausible spread of the model's performance due to sample variation in each fold [71].

Taken together, the above mentioned statistical procedures provide a multi-angle approach that strengthens the claim that any numerical advantage in F1 or accuracy is statistically grounded rather than an artifact of a single data partition or ephemeral noise.

3 Results

3.1 Embedding Configuration Comparison

As mentioned in Section 2.1, the choice of embedding approach exerted a considerable influence on how effectively the model disentangled subtle semantic cues. The GloVe embeddings successfully mapped 17,278 of the 20,000 most frequent tokens in our corpus, yielding an 86.39% coverage rate. This relatively high coverage suggests strong alignment between the academic vocabulary of COVID-19 research and the general language corpus used to train GloVe. The unmapped terms (13.61%) primarily consisted of domain-specific terminology, acronyms, and compound words particular to COVID-19 research that were absent from the general Wikipedia and Gigaword corpora.

Conversely, Word2Vec embeddings, despite being trained on a substantially larger corpus and having higher dimensionality (300 D vs. 100 D) achieved lower lexical coverage at 81.66%, representing 16,333 mapped tokens. The 4.73% percentage point difference in coverage contradicts the intuitive expectation that the larger training corpus of Word2Vec would yield higher range. This discrepancy likely stems from the nature of the Google News corpus, which, while extensive, may contain less academic and scientific terminology than the one used for GloVe and also, based on recency of the COVID-19 specific terminology.

Within these embedding approaches, trainable versus non-trainable configurations brought further nuance. Making an embedding layer adaptive permits localized fine-tuning—thereby pushing accuracy into the 97% range—but at the cost of higher variance and occasional overfitting. In contrast, keeping the same embeddings static generally stabilized validation results close to 96%, requiring fewer epochs and somewhat mitigating the risk of memorizing transient idiosyncrasies. These patterns confirmed that while trainable embeddings can tap into domain nuances, they require careful regulation through dropout or batch normalization.

3.2 Hyperparameter Tuning and Ablation Framework

3.2.1 Results of Hyperband Tuning

After meticulously exploring multiple configurations, Hyperband converged on a set of parameters that consistently improved validation accuracy while preserving robust generalization. Notably, a learning rate of 0.001 emerged as especially effective at promoting stable updates without pushing the network into oscillatory territory. Likewise, a moderate dropout of 0.1 in the recurrent layers proved sufficient for addressing overfitting, whereas the dense layer benefitted from a slightly higher dropout of 0.3. This nuanced interplay of numeric settings indicates the importance of calibrating regularization to reflect the network's size and the subtleties of textual input sequences.

Interestingly, many trials that employed a higher dropout in the LSTM layers (e.g., 0.4 or 0.5) demonstrated either slower convergence or a tendency to plateau at moderate accuracy levels, presumably due to the underutilization of the model's capacity. Conversely, extremely low dropout did not adequately control overfitting, leading to inflated training accuracy with weaker validation outcomes. The eventual "sweet spot" balanced these trade-offs and was validated across multiple runs. A further noteworthy trend emerged around LSTM layer dimensions: while 256 units in the first LSTM layer added considerable representational power, the second layer needed far fewer units—namely 32—to refine the captured features without saturating memory or training time. This separation of responsibilities between a larger, more encompassing first layer and a more compact second layer appeared to bolster generalization in the final model.

3.2.2 Ablation Findings

Ablation studies serve as powerful instruments for pinpointing which components or hyperparameters have a tangible impact on a model's predictive strength. In this investigation, two separate ablation

experiments were conducted—one in which the embedding layer remained frozen (trainable = False), and another where the embedding layer was updated during backpropagation (trainable = True). Each scenario measured (1) baseline hyperparameter sensitivity and (2) the effect of removing or altering distinct architectural components.

When the embedding layer was kept non-trainable, the model leveraged static pre-trained word vectors without adjusting them to the dataset's nuances. When the authors focus on the learning rate and batch size (see Fig. 3), the best outcomes consistently arose for a learning rate of 0.001 and smaller batch sizes (especially 32). A moderate learning rate ensured stable convergence without the rapid oscillations occasionally witnessed at 0.01. The top three parameter sets, ranked by F1 score, were {LR = 0.001, BS = 32}, {LR = 0.001, BS = 64} and {LR = 0.001, BS = 128}. Among these, an F1 of 0.9064 was the highest encountered. The ablation logs also demonstrated a striking jump in F1 (from 0.9255 to 0.9592) when the spatial dropout was removed, suggesting that spatial dropout might be superfluous or even detrimental for this text classification task in a static-embedding setting. While the "Full Model" included every recommended feature from the Hyperband search, it did not necessarily yield the highest F1 (0.9084), potentially indicating over-regularization or an excessive parameter count. In contrast, the "no_spatial_dropout" variant performed best overall, balancing depth with a more streamlined dropout approach. For the static embeddings, the ablation study indicated that and employing a learning rate of 0.001 with a batch of 32 consistently outperformed more complex alternatives.

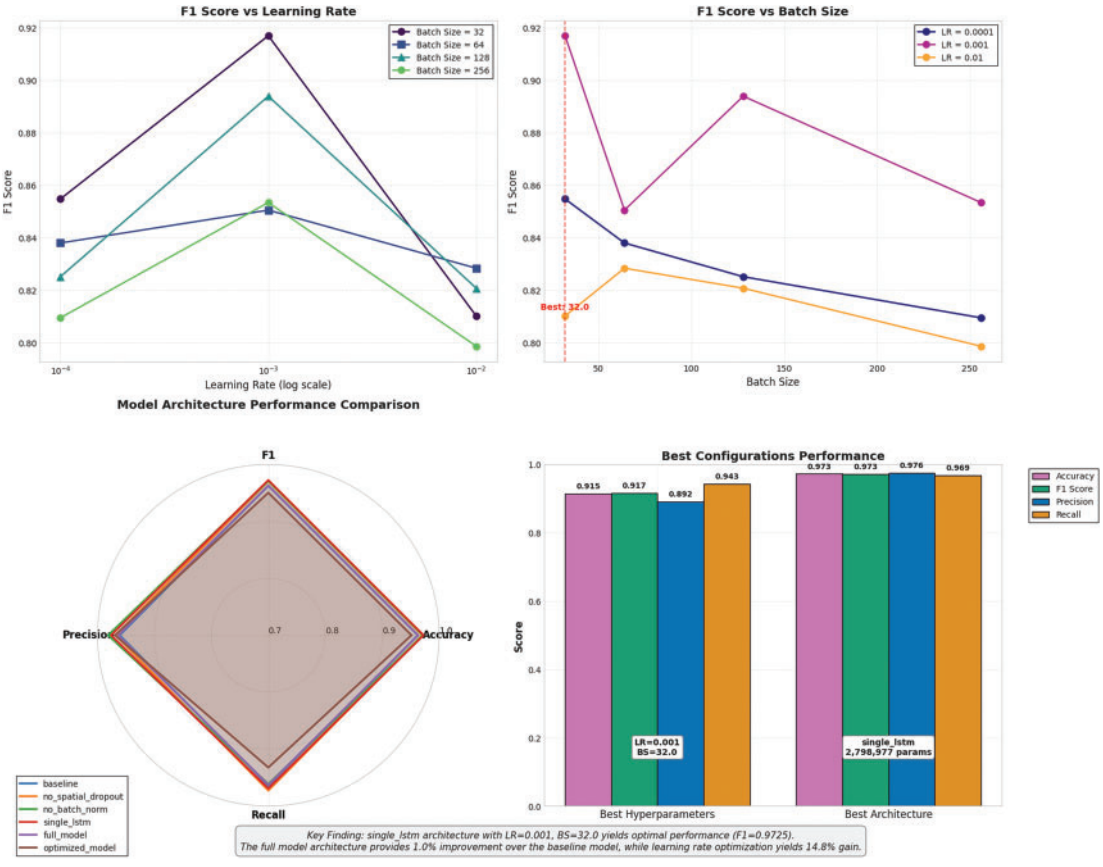

Figure 3: Impact of hyperparameters and architecture on F1 scores

Shifting to trainable embedding layers allowed the model to refine word embeddings in tandem with the classification objective, offering a more flexible feature space. According to the data produced by the

ablation study, the learning exerted a marked influence on performance. A moderate rate of 0.001, coupled with a batch of 32, once again surfaces as the optimal trade-off. Statistical analyses confirmed the statistical significance of the learning rate effects for trainable embeddings (F = 8.38, p = 0.008799), while variations in the batch size had a considerably smaller effect and were more pronounced in non-trainable settings despite not reaching statistical significance (F = 2.20, p = 0.165861). Within the range of {0.0001, 0.001, 0.01}, the synergy between a 0.001 learning rate and the smaller batch size delivered an F1 score of up to 0.9169, reinforcing the notion that granular updates at moderate scale can exploit the flexibility of trainable word embeddings.

In sharp contrast to the fixed embedding scenario, the architectural ablation demonstrated that the single bidirectional LSTM layer was the most effective configuration, slightly outperforming the models without batch normalization and without spatial dropout. This suggests that when embeddings can adapt during training, the representational capacity gained from trainable word vectors reduces the need for complex recurrent structures. The "Full Model" continued to deliver respectable accuracy, but its margin over the baseline was negligible. As a result, the simpler "single_LSTM" design took precedence, presumably avoiding the risk of overfitting that can arise when embeddings and multiple dropout layers are all learned simultaneously.

3.2.3 Model Configurations Trained

Building on the insights gleaned from ablation and tuning, a variety of model variants were ultimately trained. Table 3 outlines these configurations, which differed along three principal axes: (i) the choice of embedding source and whether it was trainable or static, (ii) the single-layer vs. dual-layer BiLSTM design, and (iii) whether standard or "optimized" hyperparameters (identified by hyperband and ablation studies) were employed. Additionally, a subset of models underwent extended training (20 epochs instead of 10) or 5-fold cross-validation to provide more robust estimates of performance generalization.

Table 3: Overview of trained BiLSTM variants

Model name/K-Fold	Embedding	Trainable?	BiLSTM layers	Batch size	Epochs
Glove Single BiLSTM (Trainable)	GloVe	Yes	Single (256 units)	32	10
Glove Single BiLSTM (Static)	GloVe	No	Single (256 units)	32	10
Word2Vec Single BiLSTM (Trainable)	Word2Vec	Yes	Single (256 units)	32	10
Word2Vec Single BiLSTM (Static)	Word2Vec	No	Single (256 units)	32	10
Glove BiLSTM (Static)	GloVe	No	Dual (256 -> 32 units)	128	10
Glove BiLSTM (Trainable)	GloVe	Yes	Dual (256 -> 32 units)	128	10
Glove BiLSTM (Optimized, Static)	GloVe	No	Dual (256 -> 32 units)	32	10
Glove BiLSTM (Optimized, Trainable)	GloVe	Yes	Dual (256 -> 32 units)	32	10
Word2Vec BiLSTM (Static)	Word2Vec	No	Dual (256 -> 32 units)	128	10

(Continued)

Table 3 (continued)

Model name/K-Fold	Embedding	Trainable?	BiLSTM layers	Batch size	Epochs
Word2Vec BiLSTM (Optimized, Static)	Word2Vec	No	Dual (256 -> 32 units)	32	10
Glove BiLSTM (Trainable, Extended)	GloVe	Yes	Dual (256 -> 32 units)	128	20
K-Fold Glove BiLSTM (Trainable)	GloVe	Yes	Dual (256 -> 32 units)	128	10
K-Fold Glove BiLSTM (Trainable, Extended)	GloVe	Yes	Dual (256 -> 32 units)	128	20
K-Fold GloVe Single BiLSTM (Trainable)	GloVe	Yes	Single (256 units)	32	10
K-Fold GloVe Single BiLSTM (Trainable, Extended)	GloVe	Yes	Single (256 units)	32	20

Each model's name reflects whether the embeddings were static (non-trainable), trainable, or if the architecture was optimized. "Single" denotes using a single BiLSTM layer with 256 units, while "Optimized" references the lack of a spatial dropout layer as per ablation study results. Extended training and cross-validation runs were carried out to more rigorously validate the stability of each model's performance.

3.3 Training Dynamics and Evaluation of the Proposed Architecture

The Full Model, as devised from the model architecture proposed and the preceding explorations, employs the dual BiLSTM layout augmented by dropout layers and batch normalization. This section analyzes the training behavior and performance characteristics of our proposed model with both static and adaptive embeddings, along with extended training experiments to determine optimal convergence properties.

3.3.1 Training Trajectories: Static vs. Adaptive GloVe Embeddings

A. Static GloVe Embeddings (trainable = False)

One core objective was to gauge the performance of our Full Model when employing GloVe word vectors as "frozen" embeddings. The reasoning is that pretrained embeddings already encode semantic relationships, thereby limiting the capacity of the network to overfit by shifting these representations to idiosyncratic traits in the training data. Fig. 4 shows the training and validation accuracies across ten epochs. After initialization, the training accuracy began near 0.7806 and climbed to 0.9466, an improvement of approximately 16.6 percentage points. By epoch 8, the model had effectively converged. As signaled by the minimal gains in validation performance thereafter. Notably, the validation accuracy at epoch 8 was 0.9629, which in fact exceeded the same epoch's training accuracy by a modest margin. This performance pattern, where validation accuracy exceeds training accuracy—represents an uncommon but theoretically grounded phenomenon in Neural Network training. The inversion of the typical accuracy relationship (where training accuracy normally exceeds validation) stems from the interaction between several model characteristics. First, the application of dropout layers affects only training-phase computation, artificially suppressing training accuracy while validation runs utilize the full network capacity. Second, batch normalization layers demonstrate

different statistical behaviors during training vs. inference, potentially favoring validation instances when the underlying data distributions have particular properties [72]. Third, the use of static embeddings constrains the model's capacity to overfit to training data, enforcing more generalized representational learning focused on sequence patterns rather than lexical memorization. The steady convergence patterns observed in both training and validation loss curves (decreasing from 0.4606 to 0.1419 for training, and 0.4523 to 0.1302 for validation) further support this interpretation. Hence, the architecture was able to isolate discriminative patterns despite lacking the flexibility to reshape word vectors.

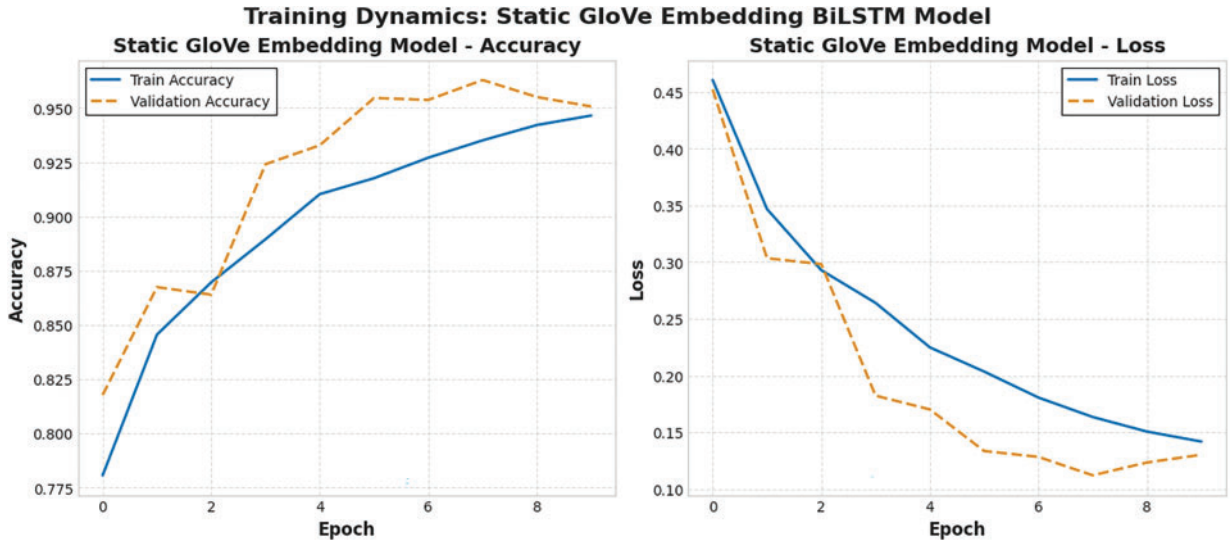

Figure 4: Training performance curves of static embedding model

B. Adaptive GloVe Embeddings (trainable = True)

Subsequently, the embedding layer was configured to be trainable, thereby enabling the model to adjust word vectors dynamically during backpropagation. This step introduces extra parameters and can, in principle, help the architecture internalize domain- or task-specific semantics that might not be comprehensively captured by the original GloVe corpus. The training curve in Fig. 5 (left panel) conveys a steeper rise than in the static scenario: the initial training accuracy of 0.7985 culminated at 0.9874 by epoch 10, an increase of nearly 18.9 percentage points. The validation accuracy, starting near 0.8395, attained 0.9725 by the final epoch, consistently trailing the training curve by about 1–2%. That relatively small gap signals moderate regularization success, especially given that the embedding space is not subject to fine-tuning. From a loss perspective, Fig. 5 (right panel), the training curve dropped sharply from 0.4265 to 0.0351, while validation loss diminished to around 0.0818. The network did exhibit slight fluctuations in validation metrics past the midpoint of training, which might foreshadow potential overfitting had the researchers continued for substantially more epochs. However, the final gap between training and validation results (about −0.0149 in accuracy) indicates that the model's capacity to overfit was tempered effectively by dropout, batch normalization, and an adaptive learning rate schedule.

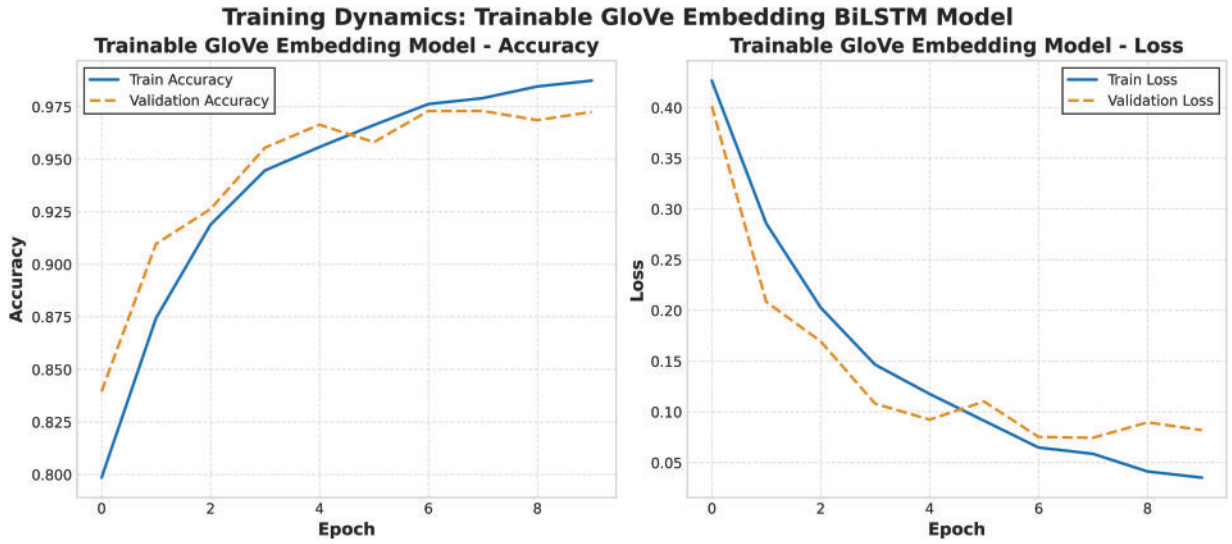

Figure 5: Training performance curves for trainable GloVe model

3.3.2 Model Classification Results

A. Static GloVe Model

On the test set (5732 samples evenly split between original and AI-generated abstracts), the final static model achieved 0.9513 accuracy and an F1 score of 0.9508 (see Table 4). From the confusion matrix, it is observed that 2758 of the original abstracts were correctly identified (true negatives), with only 108 false positives. Similarly, the model labeled AI-generated texts accurately 2695 times, with 171 false negatives, as can be seen in Table 4. The resultant recall for AI-generated detection was 0.9403, meaning that while performance is robust, the network occasionally struggles to detect certain AI-specific lexical cues. The precision score of 0.9615 indicates that when the model classified a text as AI-generated, it was correct in the majority of cases, reflecting a high degree of reliability in its positive classifications.

Table 4: Evaluation metrics for static and trainable GloVe configurations

Metric	Static GloVe model	Trainable GloVe model
Accuracy	0.9513	0.9698
F1 Score	0.9508	0.9701
Precision	0.9615	0.9622
Recall	0.9403	0.9780
Specificity	0.9623	0.9616
True negatives (correctly identified Original)	2758	2756
False positives (Original misclassified as AI-Generated)	108	110
False negatives (AI-Generated misclassified as Original)	171	63
True positives (correctly identified AI-Generated)	2695	2803
False positive rate	0.0377	0.0384
False negative rate	0.0597	0.0220
Overall error rate	0.0487	0.0302

Interestingly, the false-negative rate (~5.97%) surpassed the false-positive rate (~3.77%). In tasks that prioritize detecting AI-generated content (e.g., editorial workflows or plagiarism detection), such a difference can be critical: missing real AI-generated texts (false negatives) may pose a bigger concern than occasionally misclassifying an authentic human-written document. Nonetheless, the overall error of 4.87% is fairly modest, reflecting that the static embedding approach in a well-regularized BiLSTM can deliver strong generalization.

B. Trainable GloVe Model

Upon evaluation, the trainable-embedding version of the Full Model reached 0.9698 accuracy and an F1 score of 0.9701. The slight edge over the static variant is most visible in recall, which rose from 0.9403 to 0.9780. In other words, the model with adaptive embeddings caught more of the AI-generated texts that the previous approach missed. The cost is a minor dip in specificity (from 0.9623 to 0.9616), meaning it occasionally re-labeled real abstracts as AI-generated. For tasks where identifying AI-generated content is paramount, a higher recall might be preferable.

Comparing the two configurations shows that trainable embeddings yield about 1.95%–2.05% absolute improvement in key metrics (accuracy, F1) but at the expense of increased sensitivity to training epochs and possible overfitting. Despite the great complexity, the model converged around epoch 8–9, similar to the static version. The end result is a noticeable improvement in capturing AI-specific attributes, signaled by a ~38% reduction in overall error rate.

3.3.3 Longer Training Regime Analysis

Prior experiments concluded around epochs 8–10, frequently due to the activation of early stopping criteria. However, it was hypothesized that extending the training duration might yield further performance improvements or, conversely, lead to significant overfitting. To investigate this potential, the trainable-embedding Full Model was allocated a training budget of up to 20 epochs, with intermediate metrics such as precision, recall, and F1 score monitored throughout the training process.

Fig. 6 provides a detailed visual overview of the model's extended training, showing how accuracy, loss, precision, recall, and F1-score changed over time. It also includes the final confusion matrix, confidence intervals calculated using bootstrapping, and the time taken for each training cycle (epoch). The model underwent a total of 15 training epochs but implemented an early stopping mechanism, terminating training when the validation loss plateaued and subsequently exhibited a marginal increase. A cursory glance at the accuracy and F1 charts demonstrates that most improvements occurred before epoch 10, with marginal gains at epochs 8–9. Notably, the best validation accuracy was 0.97 at epoch 8—remarkably close to the standard 10-epoch schedule used previously. The best validation loss was found as well at epoch 8, the subsequent epochs offering negligible advantage or a mild regression. Compared to the earlier 10-epoch run with trainable embeddings, researchers observed that extended training neither harmed performance drastically nor delivered major leaps forward. Rather, it refined certain aspects (like a narrower difference between training and validation) but left the final evaluation metrics roughly in the same ballpark.

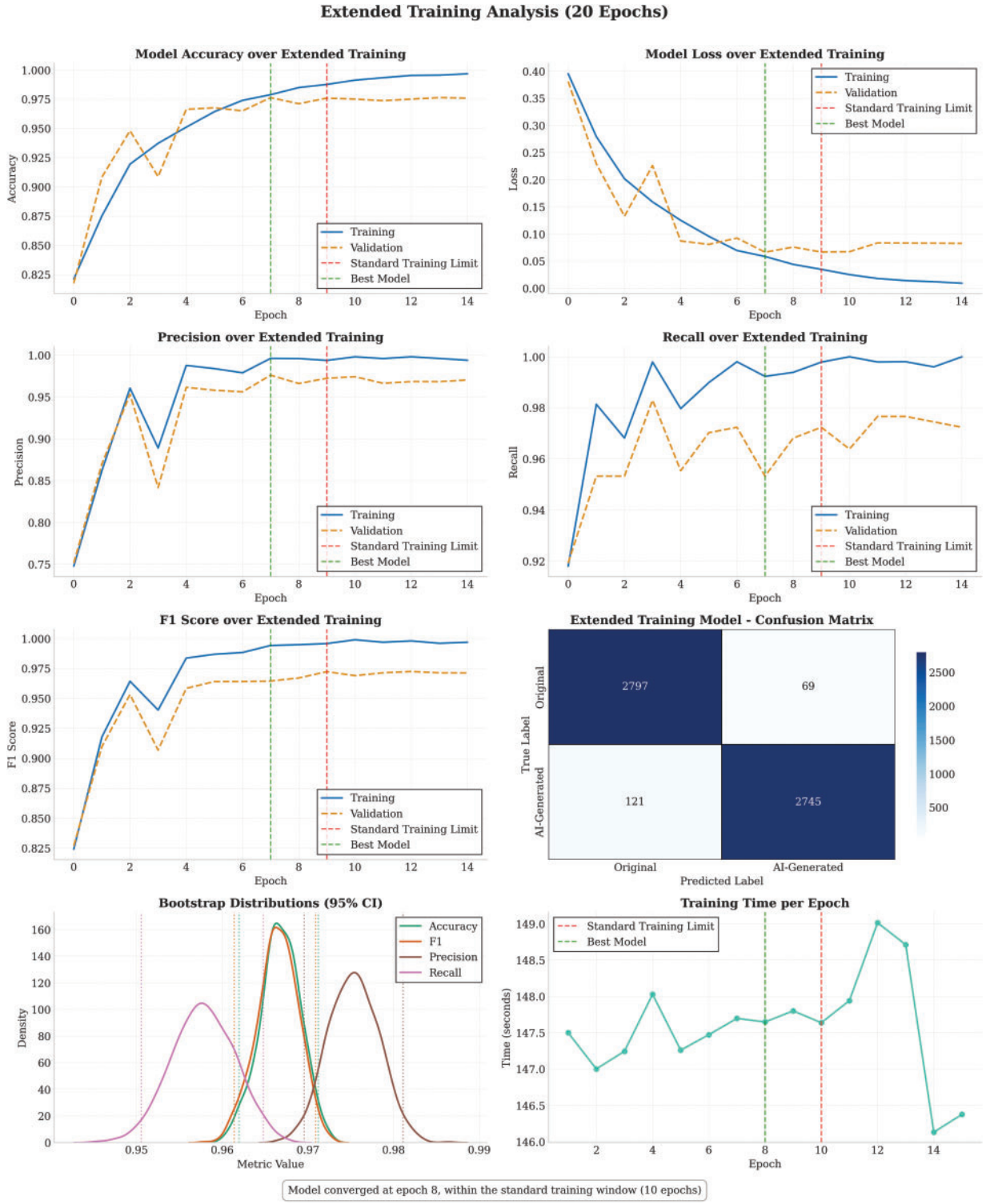

Figure 6: Prolonged training evaluation

To assess the stability and reliability of the model's final predictions, a bootstrap resampling procedure comprising 1000 iterations was conducted. The 95% confidence interval for accuracy ranged from 0.9620

to 0.9712. Similar intervals for F1 spanned 0.9614 to 0.9709, with narrower bounds on precision (0.9696–0.9811) and a slightly wider range for recall (0.9505–0.9648). Thus, any random partitioning variations in the test set are unlikely to produce major shifts beyond a fraction of a percent, signifying that the model's performance is fairly stable. From an operational standpoint, extended training added about 5 more epochs beyond the usual stopping point, consuming an extra 12–13 min of GPU time for an overall 39.97-min run. The final "best" checkpoint still occurred at epoch 8, so the additional epochs were effectively superfluous. An improvement of only around +0.04% in validation accuracy was detected from epoch 10 to epoch 9—well within the margin of random fluctuations. Hence, while a maximum of 20 epochs gave the model an opportunity to search for superior solutions, early stopping rendered those additional iterations moot.

Though the trainable embeddings model performed slightly better in terms of raw metrics, its pattern of validation loss and the need for careful interpretation cannot be overlooked. Both models approached, but did not surpass, earlier benchmarks set by LSTM-Word2Vec research [9]. More critically, neither configuration approached 100% accuracy, reminding us that even refined neural architectures cannot unequivocally guarantee flawless detection.

3.4 K-Fold Cross-Validation Analysis

The generalization capabilities were thoroughly investigated using a rigorous 5-fold stratified Cross-Validation (CV) procedure. The stratification ensured an equal representation of both classes (human and AI-generated) across all partitions, preserving the balanced distribution present in the dataset comprising 28,662 abstracts equally divided between the two classes. Each fold consisted of approximately 5732 abstracts, maintaining the class ratio precisely at 50% per category.

The cross-validation experiments were conducted on four model configurations designed to elucidate the effects of architectural complexity (single-layer vs. dual-layer BiLSTM) and training duration (standard vs. extended training epochs). Specifically, the models under scrutiny were: (1) Single-layer BiLSTM trained for 10 epochs; (2) Single-layer BiLSTM trained for 20 epochs (extended training); (3) Dual-layer BiLSTM (trainable GloVe embeddings) trained for 10 epochs; (4) Dual-layer BiLSTM (trainable GloVE embeddings) trained for 20 epochs (extended training).

3.4.1 K-Fold Model Convergence and Accuracy Profiles

Fig. 7 displays two panels that track the validation accuracy and validation loss at each training epoch, averaged across all folds for each model. A detailed observation of the validation accuracy plot reveals a generally rapid convergence trajectory across all configurations, with notable increments within the initial three to four epochs, suggesting that the neural architectures quickly adapted to capture the main features distinguishing human from AI-generated abstracts. Particularly, the single-layer BiLSTM models, irrespective of epoch count, display the steepest ascent initially, achieving higher accuracy earlier relative to dual-layer counterparts.

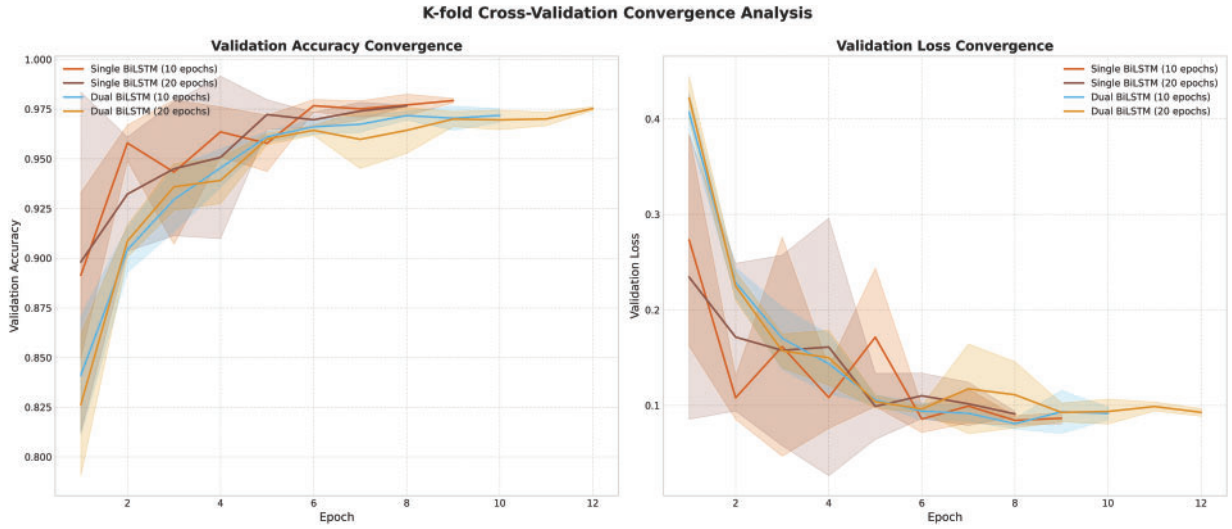

Figure 7: K-fold cross-validation convergence analysis

Turning attention to the validation loss convergence, the models uniformly illustrate sharp decreases within the initial epochs, rapidly reaching a plateau phase between epochs four and six. Single BiLSTM (10 epochs) shows substantial fluctuation in loss during intermediate epochs, indicating higher volatility in optimization, likely due to stochastic variations in data subsets. However, it eventually stabilizes, aligning closely with the other configurations at the final stages. Both dual-layer architectures show a consistent and uniform reduction in loss, suggesting more predictable convergence dynamics, potentially due to the regularizing influence of increased network complexity and parametrization. Paired statistical tests, performed on the fold-level metrics, reveal that these extended vs. standard training differences typically lack significance (p-values > 0.05), reinforcing that simply running more epochs does not guarantee a tangible jump, therefore proving empirical evidence that prolonged training contributes negligible benefit to model generalization capacity.

3.4.2 Fold-Specific Performance Distribution and Comparison

Fig. 8 provides a detailed analysis of the F1-scores obtained across the different folds of the cross-validation procedure (top panels). Furthermore, it aggregates these results into distribution-based comparisons (bottom panels), enabling a thorough evaluation of each model's performance consistency and robustness when exposed to diverse partitions of the dataset.

Across folds 1–5, the single layer architecture trained for 20 epochs attains the highest mean F1 (0.9745), accompanied by a low coefficient of variation (~0.17%), reflecting very consistent performance. In contrast, the extended dual-layer BiLSTM exhibits more pronounced variability (~0.53 CV), including occasional dips in a certain fold. Further statistical scrutiny via t-tests on fold-level F1 revealed that the minor improvements afforded by 20-epoch training in the single-layer design were not large enough to reach significance (p = 0.4000). Equally, the dual-layer's slight decline under extended training was not statistically meaningful (p = 0.6681). However, contrasting the single-layer model with the dual-layer model did uncover a statistical advantage for the former (p = 0.0155), implying that additional architecture complexity did not translate into performance improvements commensurate with its computational costs.

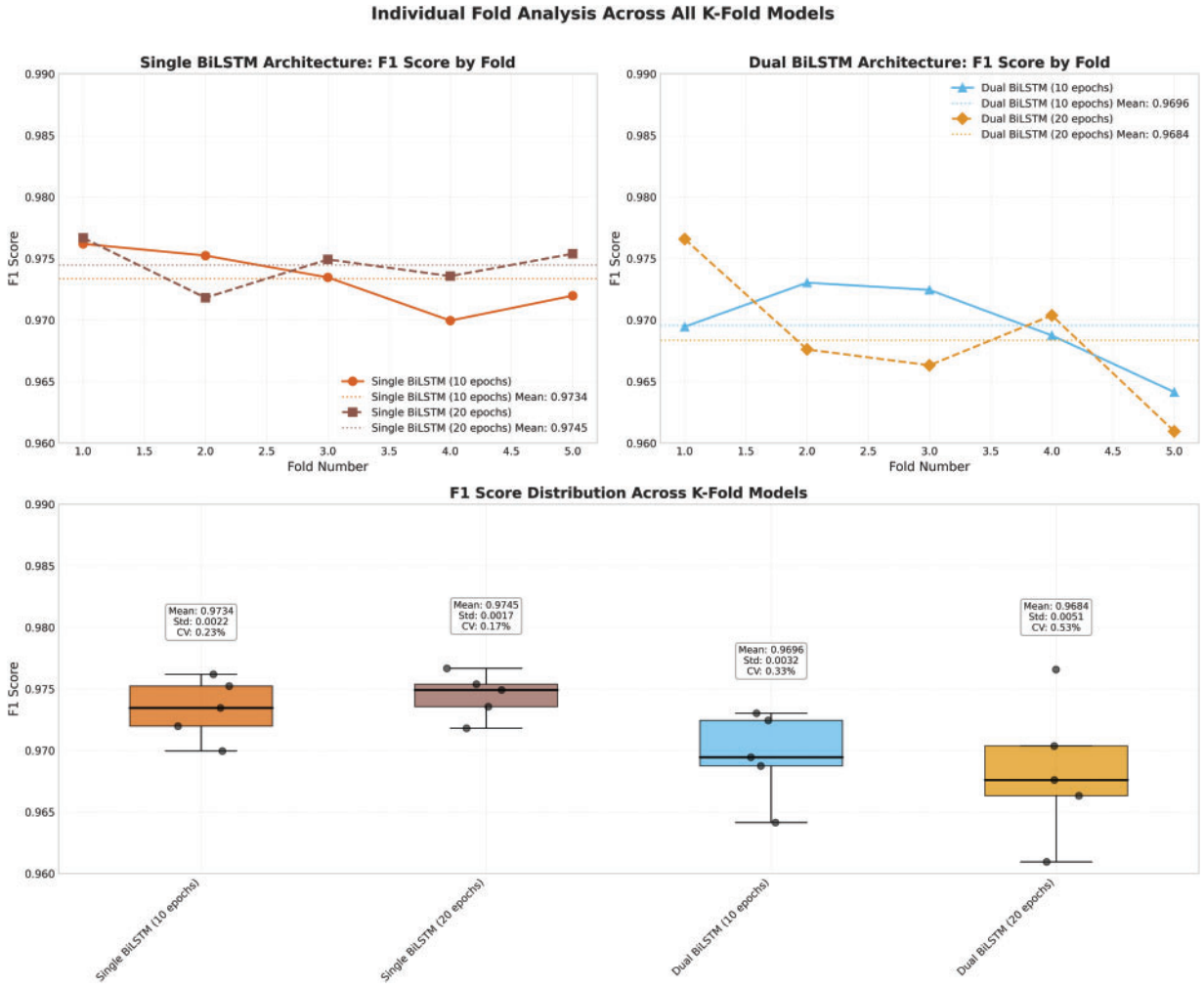

Figure 8: Individual fold analysis across all K-fold models

Taken together, the results from the cross-validation validate the core observations from single-split experiments by providing substantive evidence supporting architectural simplification in this specific text classification task. The combination of superior absolute performance, enhanced stability across data partitions, and reduced parameter count renders the single-layer BiLSTM architecture clearly preferable to the dual-layer configuration.

3.5 Cross-Architecture Performance Review

In an effort to contextualize the BiLSTM architectures detailed in Section 3.2.3, the authors expanded the comparisons to include two additional neural baselines from Section 2.5: a CNN with GloVe embeddings and unidirectional LSTM with GloVe embeddings. By juxtaposing these simpler DL setups with our more evolved bidirectional models, the authors aimed to determine whether the added complexity of bidirectionality or refined hyperparameter tuning genuinely translated into substantive gains.

3.5.1 Comparison of Neural Architectures and Embeddings

Fig. 9 presents a comparative analysis of classification performance across multiple models, showcasing their respective accuracy, F1-score, precision, and recall metrics. The high-level takeaway is that BiLSTM architectures consistently outperform both the CNN and the unidirectional LSTM on nearly every metric, thereby providing evidence that the capacity to integrate context from both preceding and subsequent tokens substantially enhances the model's ability to detect subtle attributes of AI-generated text. Even the simpler single-layer BiLSTM variants, operating with moderate hyperparameter tuning, noticeably surpass the best performances recorded by the CNN or unidirectional LSTM. In particular, the gap in F1 often hovers between 0.02 and 0.05, a difference that repeated significance tests (including McNemar's test on classification disagreements) confirm is unlikely due to random fluctuation.

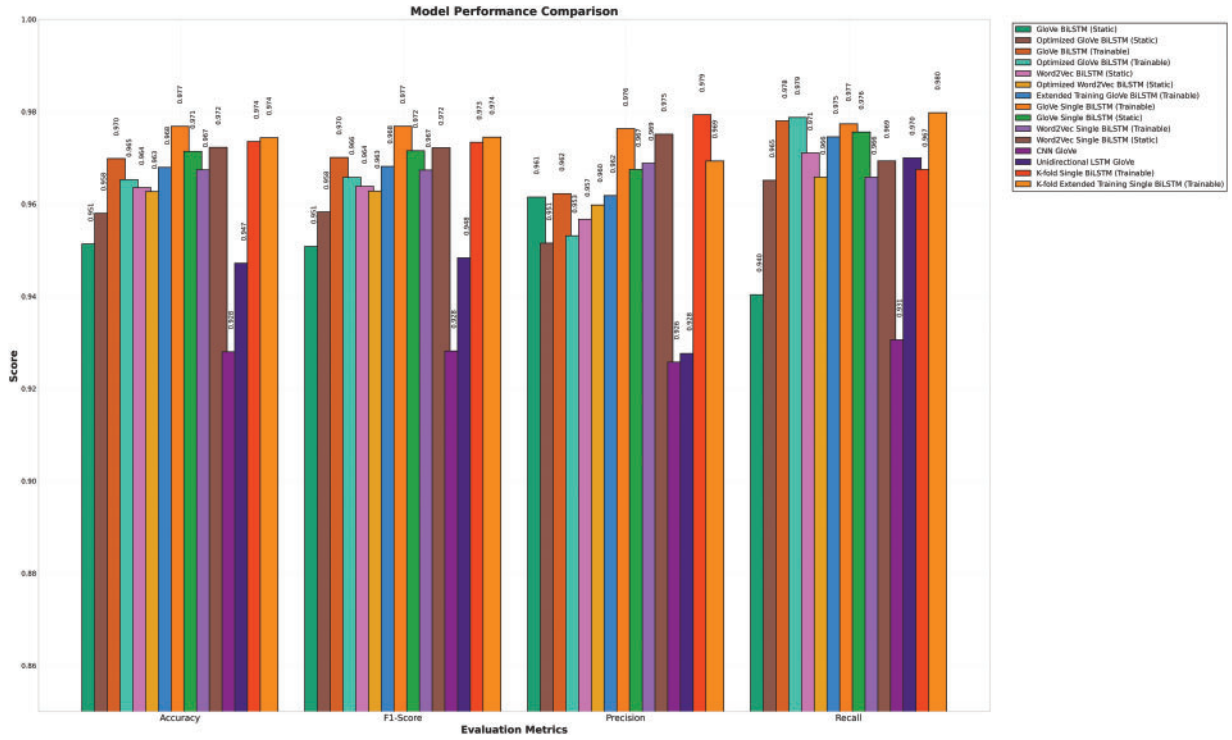

Figure 9: Multi-model classification metrics

In parallel, a close look at the embedding source reveals that GloVe-based BiLSTMs generally reach or exceed 0.96 F1, with top contenders pushing beyond 0.97 or even 0.98. Word2Vec-based models occasionally cross the 0.96 threshold—especially with static embeddings—but more commonly settle around 0.95–0.96. This moderate shortfall can be traced to the coverage disparities noted earlier, when GloVe's pretraining corpus appears to align more closely with academic vocabulary, giving it a slight edge on domain terms and nuanced phrasing typical of scientific abstracts. Nonetheless, Word2Vec remains a valid choice in contexts where the end goal tolerates a small performance deficit for some other advantage (e.g., if Word2Vec embeddings are already in use for a multi-task pipeline). Certain Word2Vec models do, in fact, outperform suboptimal GloVe configurations, underscoring that well-tuned Word2Vec systems are still broadly competitive if carefully managed.

Multiple "optimized" versions—broadly referencing the ablation-driven removal or reduction of spatial dropout, along with a few additional hyperparameter tweaks—demonstrate that these targeted adjustments can sometimes yield a modest boost. For instance, "Optimized GloVe BiLSTM (Static)" achieves about +0.7 percentage points in accuracy relative to the non-optimized static counterpart, largely attributable to removing spatial dropout in a scenario where overfitting was less of a concern. However, the story is not universal: "Optimized GloVe BiLSTM (Trainable)" falls slightly below the standard trainable GloVe BiLSTM, suggesting that ablation insights like removing certain dropout layers do not always produce net benefits once the embeddings themselves are allowed to adapt. The same pattern recurs in Word2Vec: "optimized" sometimes yields better performance, other times not. From a practical standpoint, these results imply that optimization measures directed by ablation studies can help in static-embedding contexts (where the model has fewer trainable parameters to regulate) but may prove less advantageous or even detrimental when embeddings are also being updated.

Ultimately, the findings consistently reaffirm that single-layer BiLSTM architectures, particularly those comprising 256 hidden units, constitute a robust and highly performant configuration. Some single-layer GloVe models comfortably surpass dual-layer versions by 1–2 percentage points in F1 or recall. This disparity emerges most clearly in the cross-validation fold-level data, where single-layer networks appear to converge more robustly and exhibit fewer fluctuations across folds than their dual-layer counterparts. Significance tests, including paired t-tests on fold metrics, occasionally yield p-values around 0.015, suggesting the differences are meaningful and not mere sampling noise. With Word2Vec, a second layer might sometimes help mitigate coverage shortfalls, but the best single-layer GloVe configurations are rarely eclipsed.

From a deployment perspective, the findings underscore that extending a BiLSTM architecture to two layers or incorporating multiple optimizations does not inherently yield a net performance benefit. A carefully tuned single BiLSTM with GloVe embeddings—preferably trainable if one can manage the risk of overfitting—tends to furnish the highest F1 scores while maintaining strong recall. Indeed, these top configurations markedly outdistance simpler neural baselines (CNN or unidirectional LSTM), verifying that bidirectionality and judicious hyperparameter decisions can yield real improvements in detecting AI-generated text.

3.5.2 Statistical Perspective on Model Comparisons

The quantitative differences observed across the various model configurations discussed in the preceding subsections—including BiLSTM architectures, single-split vs. cross-validation data partitioning, and diverse embedding strategies—warrant a systematic statistical analysis. This study employed a variety of statistical procedures—most prominently McNemar's test for pairwise classification comparisons, plus certain paired t-tests on per-fold metrics in the cross-validation context—to approximate whether performance differentials are stable across sampling perturbations. The impetus behind these tests is to avoid over-reliance on a single-split metric that might unintentionally favor one model's luck with a particular partition.

A. McNemar's Test for Paired Classifications

The analysis identified multiple statistically significant distinctions among the model variants. The Single BiLSTM with trainable GloVe embeddings emerged as statistically superior to multiple competing architectures, registering 16 significant victories in pairwise comparisons. This robust statistical performance stands in marked contrast to the CNN-GloVe architecture, which demonstrated statistical superiority in only four pairwise comparisons. Several notable patterns emerged from the McNemar analysis (see Fig. 10). First, architectural complexity did not necessarily correlate with statistical superiority—the simpler Single BiLSTM architectures frequently outperformed their more complex dual-layer counterparts. Second, embedding

adaptability proved statistically advantageous, with trainable embedding configurations consistently outperforming their static counterparts. Third, Word2Vec-based models demonstrated competitive statistical performance despite their lower absolute metrics, suggesting effective feature extraction from their higher-dimensional embedding space. Particularly notable was the absence of statistically significant differences between certain model pairs. The Trainable GloVe BiLSTM and the Extended Training BiLSTM variant showed statistically indistinguishable performance ($p = 0.3823$), suggesting that additional training epochs provided minimal classification advantage beyond standard training duration. Similarly, Single BiLSTM with GloVe static embeddings and trainable GloVe BiLSTM demonstrated statistically equivalent performance ($p = 0.5585$), indicating that architectural simplification and embedding trainability may represent alternative pathways to similar performance levels.

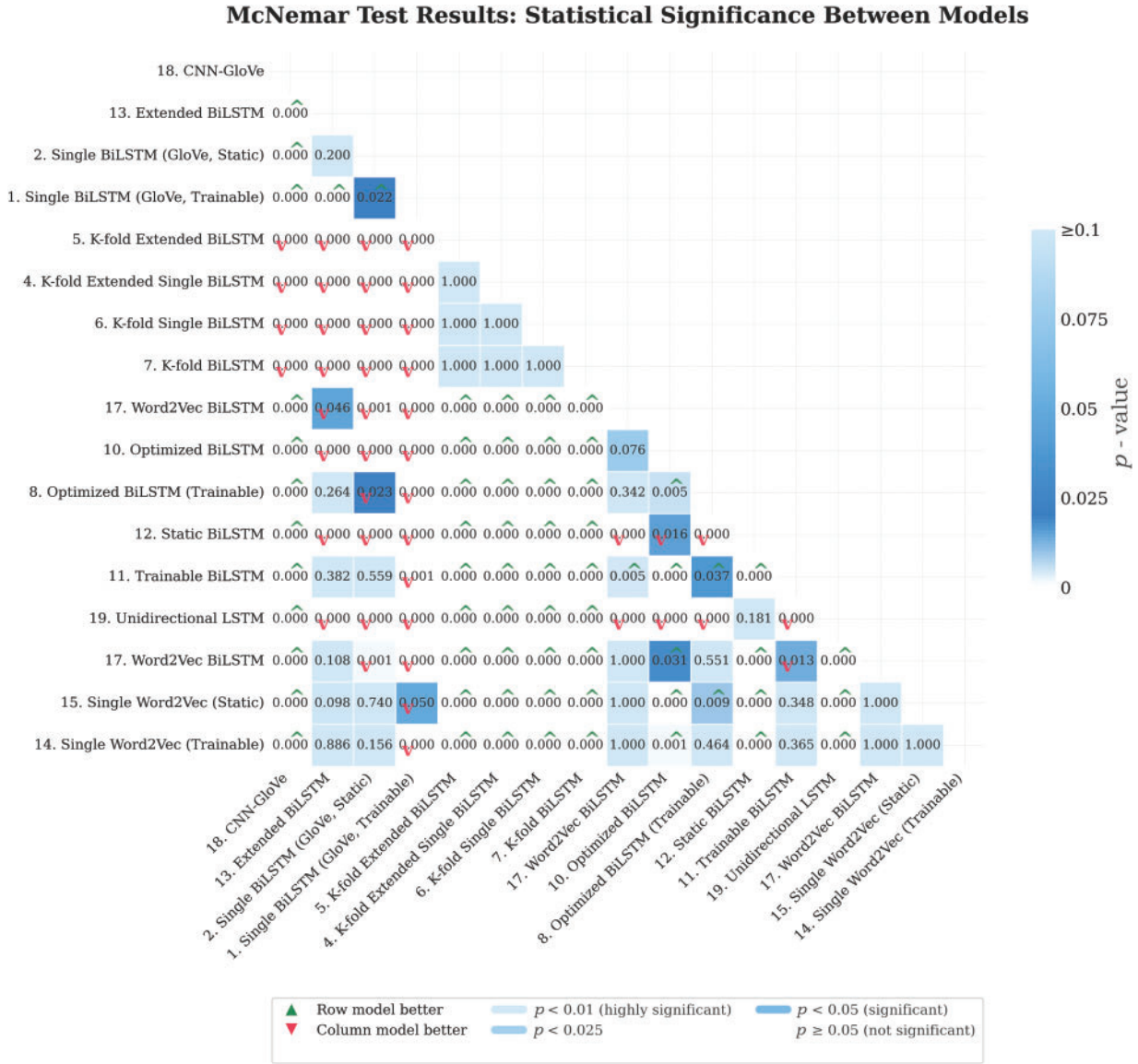

Figure 10: Matrix of McNemar's test outcomes for model pairs

B. Practical Significance via Odds Ratios and Confidence Intervals

While statistical significance establishes the existence of performance differences, effect size quantifies their magnitude and practical relevance. The authors calculated odds ratios and their corresponding 95% confidence intervals for each pairwise comparison to assess practical significance. Fig. 11 presents these effect sizes as a forest plot for the most noteworthy comparisons.

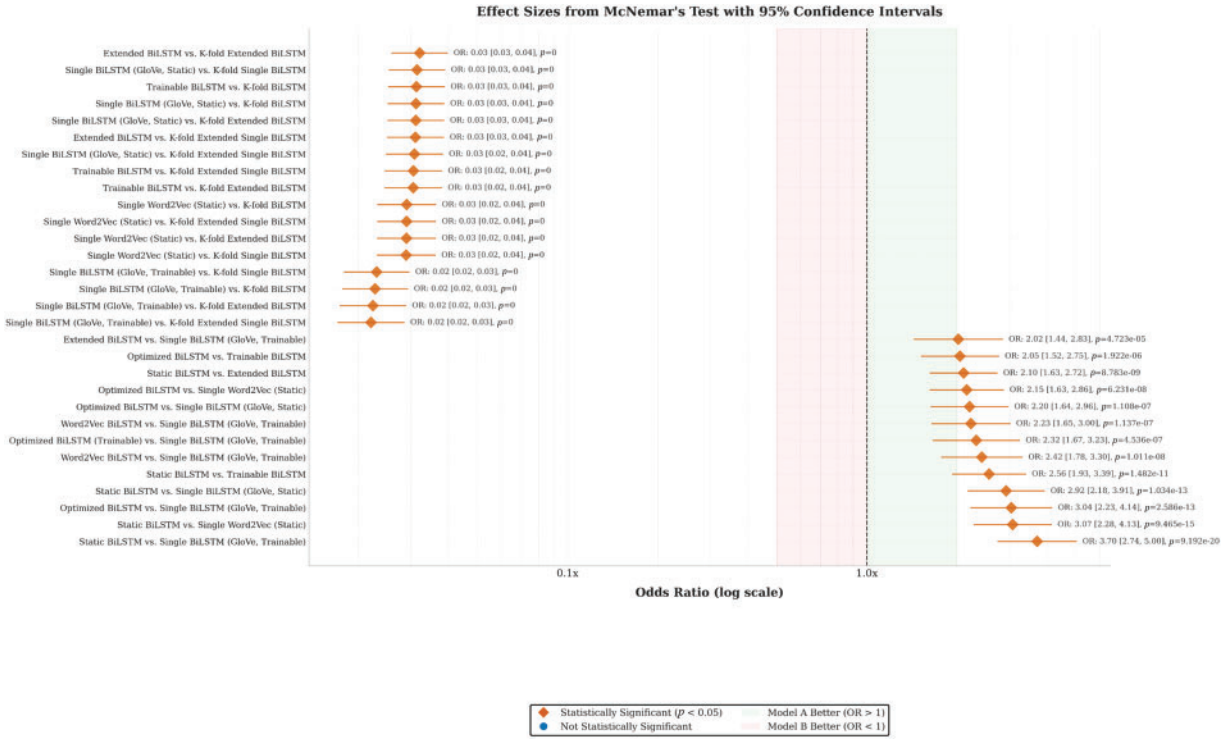

Figure 11: Effect sizes from McNemar's test with 95% confidence intervals

The effect size analysis revealed considerable practical differences between specific pairs of models. The most pronounced effect emerged in the comparison between Static BiLSTM and Single BiLSTM with trainable GloVe embeddings (OR = 3.704, 95% CI: [2.742,5.002]), indicating that the odds of correct classification by the Single BiLSTM with trainable embeddings were approximately 3.7 times higher than for the Static BiLSTM when the models disagreed. Similarly, the Optimized BiLSTM demonstrated substantially higher odds of correct classification compared to Single BiLSTM with trainable GloVe (OR = 3.038, 95% CI: [2.227,4.143]). Intriguingly, the largest effect sizes generally occurred in comparisons involving Single BiLSTM architectures, reinforcing the finding that architectural simplification effectively enhanced classification performance. This suggests that model complexity reduction not only simplified the architecture but substantially improved its ability to correctly classify instances where more complex models failed. Narrower intervals, such as those for comparisons involving extensively validated models, indicate more precise effect size estimation and greater confidence in the practical significance of observed differences.

C. Additional Validation Methods and Significance Tests

In addition to McNemar's test, paired t-tests were conducted across cross-validation folds to further assess statistical significance. In that scenario, each of the five folds yields a performance figure—accuracy, F1, or some other measure—for each model, thereby creating matched pairs across folds. The t-test on these paired results can detect whether the average difference in F1 (for instance) is likely to be zero or not. The text reveals that certain comparisons, such as single vs. dual BiLSTM for standard training, do not show

significance, while single-layer extended training vs. single-layer standard training often yields *p*-values well above 0.05, indicating no robust difference. Notably, occasionally yields borderline statistical significance for isolated metrics—such as recall or precision—yet these effects are inconsistent and do not reliably generalize across other performance across other metrics or data folds. The upshot is that extended training presumably helps in some cases but remains far from mandatory, especially when the differences remain within the noise margin.

Summarizing across all significance tests, three principal conclusions about the deep models stand out: (1) numerous advanced BiLSTM configurations significantly outperformed the baseline CNN and unidirectional LSTM systems; (2) single-layer BiLSTMs with trainable GloVe embeddings exhibit either a marked advantage or, at minimum, performance parity relative to deeper or ostensibly optimized variants—with several comparisons yielding statistically significant results favoring the simpler architecture; and (3) Word2Vec models can keep up in certain contexts but seldom dethrone the top GloVe systems. These overarching patterns echo the raw, metric-based ranking presented in earlier sections; however, the application of statistical analysis ensures that performance differentials are interpreted through the framework of classification disagreements and repeated fold-based validation, rather than relying solely on single-split evaluation snapshots.

3.5.3 Contrasting Advanced Neural Architectures with Classic ML

In tandem with the neural comparisons, the study also conducted a comparative benchmark to evaluate the performance of modern BiLSTM architectures relative to more traditional classification pipelines. These classical pipelines included linear classifiers (such as logistic regression) and ensemble approaches (e.g., random forests, gradient boosting, and Naïve Bayes), each appended to a TF-IDF or Bag-of-Words vectorizer for textual feature extraction. Although earlier research has shown these methods can be durable and transparent, the question was whether they could match or surpass the best BiLSTM solutions.

Upon evaluating TF-IDF-based classifiers (see Table 5), it is evident that a well-configured pipeline (often featuring XGBoost or logistic regression at the end) can reliably surpass older DL baselines, including simpler CNN models or unidirectional LSTM structures. In fact, the top-tier classical methods approached or exceeded F1 scores around 0.960—an achievement that was once unattainable for older RNN-based systems in earlier studies of textual classification. This evidently indicates that if one's sole comparison were Naïve Bayes (hovering near 0.86–0.88 F1) or a more primitive neural architecture, these classical approaches might appear quite compelling. However, they generally did not keep pace with a refined single-layer trainable BiLSTM, which repeatedly climbed toward 0.97 or 0.98 in F1, thus constituting a difference that was not merely random but shown to be statistically tangible in the McNemar analyses. The underlying reasons for this disparity between advanced neural approaches and traditional pipelines likely stem from multiple factors. One key dimension is that a BiLSTM with pre-trained embeddings can exploit both local sequential structure and deeper semantic similarities among words. Traditional ML systems dependent on TF-IDF or Bag-of-Words operate as purely lexical-level scorers, weighting tokens that appear frequently in certain classes. While they benefit from simpler optimization and clarity in interpretability (weights can be inspected to see which n-grams the model favors), they can falter when confronted by synonyms or domain-specific terms that do not manifest in the training distribution. In an AI detection context, LLMs frequently produce text that is not simply a rehash of training set tokens, but a more cunning rearrangement or paraphrase. The capacity of neural embeddings to cluster semantically adjacent vocabulary presumably empowers them to generalize beyond the discrete n-grams that classical methods require. Furthermore, in tasks that revolve around subtle textual cues—such as the slightly off-kilter transitions or certain lexical-linguistic anomalies that hint at an AI-generated passage—BiLSTMs harness memory cells to integrate information across

sentence boundaries. In contrast, traditional pipelines weigh each n-gram (or collection of words in a limited window) almost independently, failing to capture the ephemeral but potentially revealing transitions in style or syntax. As evidenced, while XGBoost or logistic regression might do a respectable job of capturing highly discriminative phrases, they can be outmaneuvered by the flexible gating mechanisms of LSTMs, especially when these gates adapt to new forms of AI text that do not match any previously seen lexical pattern.

Table 5: Ranked F1 model performances across methods

Rank	Model	Category	F1 score	Accuracy	Precision	Recall
1	GloVe Single BiLSTM (Trainable)	Single Layer BiLSTM	0.9768	0.9768	0.9763	0.9773
2	K-fold Extended Training Single BiLSTM	Single Layer BiLSTM	0.9745	0.9743	0.9693	0.9798
3	K-fold Single BiLSTM (Trainable)	Single Layer BiLSTM	0.9734	0.9735	0.9793	0.9675
4	Word2Vec Single BiLSTM (Static)	Single Layer BiLSTM	0.9722	0.9723	0.9751	0.9693
5	GloVe Single BiLSTM (Static)	Single Layer BiLSTM	0.9715	0.9714	0.9675	0.9756
6	GloVe BiLSTM (Trainable)	Dual Layer BiLSTM	0.9701	0.9698	0.9622	0.9780
7	Extended Training GloVe BiLSTM (Trainable)	Dual Layer BiLSTM	0.9681	0.9679	0.9618	0.9745
8	Word2Vec Single BiLSTM (Trainable)	Single Layer BiLSTM	0.9673	0.9674	0.9688	0.9658
9	Optimized GloVe BiLSTM (Trainable)	Dual Layer BiLSTM	0.9657	0.9653	0.9531	0.9787
10	Word2Vec BiLSTM (Static)	Dual Layer BiLSTM	0.9638	0.9635	0.9567	0.9710
11	TF-IDF + Logistic Regression	Traditional ML	0.9630	0.9634	0.9719	0.9543
12	Optimized Word2Vec BiLSTM (Static)	Dual Layer BiLSTM	0.9628	0.9627	0.9598	0.9658
13	TF-IDF + XGBoost	Traditional ML	0.9602	0.9602	0.9618	0.9585
14	Optimized GloVe BiLSTM (Static)	Dual Layer BiLSTM	0.9583	0.9580	0.9515	0.9651
15	GloVe BiLSTM (Static)	Dual Layer BiLSTM	0.9508	0.9513	0.9615	0.9403
16	Unidirectional LSTM GloVe	DL Baseline	0.9483	0.9471	0.9276	0.9700
17	TF-IDF + Random Forest	Traditional ML	0.9404	0.9386	0.9129	0.9696
18	TF-IDF + Gradient Boosting	Traditional ML	0.9315	0.9328	0.9503	0.9135
19	CNN GloVe	DL Baseline	0.9281	0.9279	0.9257	0.9306
20	BoW + Naïve Bayes	Traditional ML	0.8587	0.8538	0.8307	0.8887

Despite their lower ultimate ceiling, classical methods remain relevant in situations where immediate interpretability or minimal computational overhead outrank the quest for the absolute highest F1. An organization with strict resource limitations, or one that must produce a transparent record of which token-level features swayed the decision, might favor a logistic regression pipeline. Clearly, logistic regression or random forest can be trained in a fraction of the time that a BiLSTM typically requires, circumventing repeated epochs of forward and backward passes through a large embedding space. Consequently, users may be inclined to accept an F1 score in the vicinity of 0.96, as opposed to 0.97 or 0.98, particularly when such a marginal reduction in accuracy is considered acceptable within production settings characterized by frequent model retraining or the processing of exceptionally large text corpora.

The category-level analysis presented in Fig. 12, which aggregates average F1-scores and accuracy, provides further empirical support for these findings. Traditional ML methods, as a broad category, achieve an average F1 in the lower 0.93 range, overshadowing older neural baselines (such as unidirectional LSTM or CNN with an F1 around 0.928–0.948) but lagging behind modern single-layer BiLSTMs, which average above 0.97. The presence of a small standard deviation among the top ML models indicates that some approaches, like XGBoost or logistic regression with TF-IDF, push near 0.96 or 0.963, but none appears poised to breach the 0.97 threshold that single-layer GloVe accomplishes. The synergy between coverage, gating dynamics, and learned embeddings evidently fosters a margin that simpler pipelines cannot close.

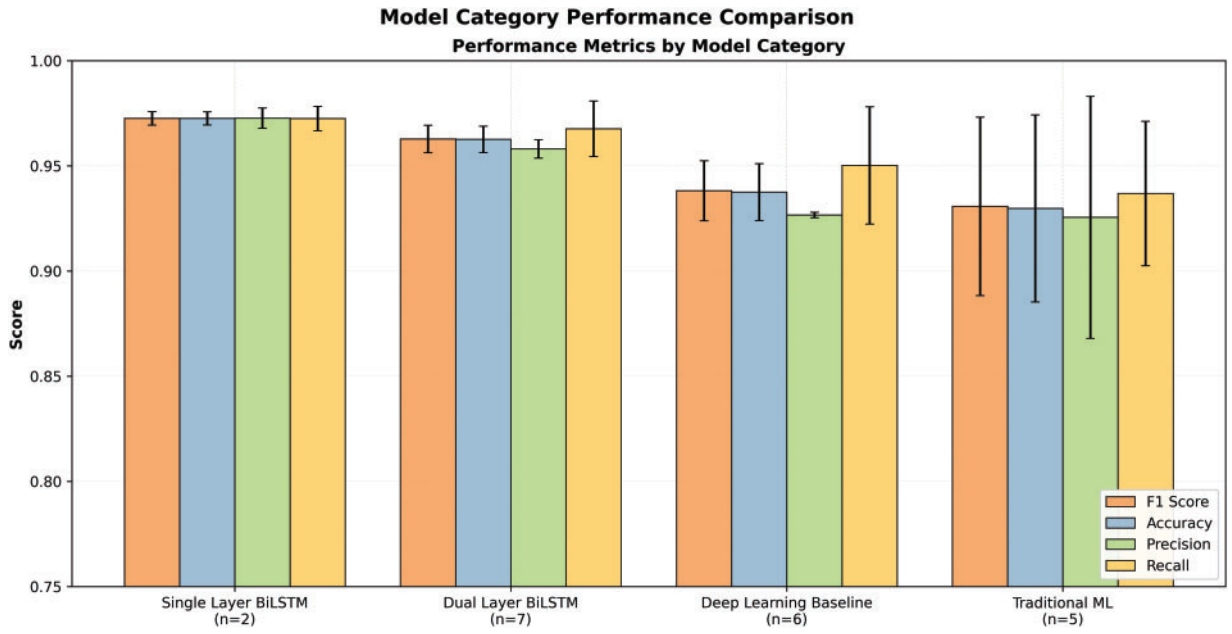

Figure 12: (Continued)

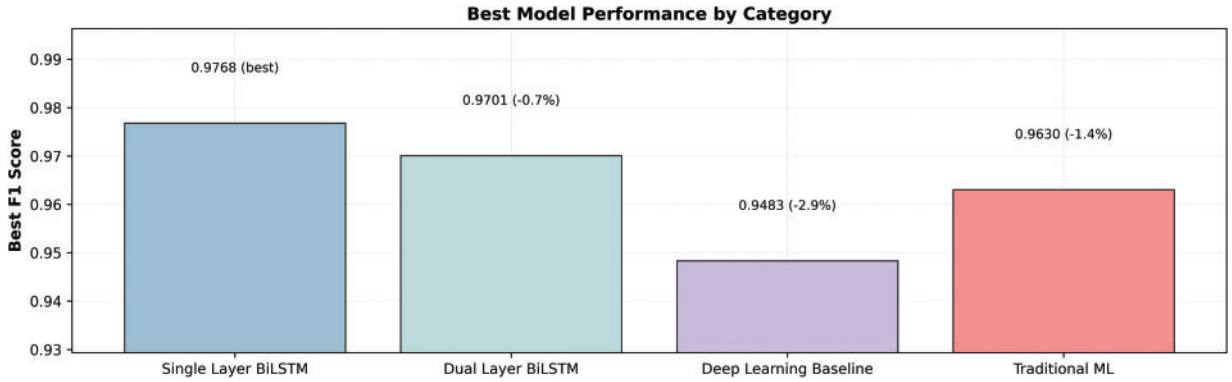

Figure 12: Category-wise performance overview (neural vs. traditional)

4 Discussion

The discourse surrounding the detection of AI-generated text has reached a critical juncture, especially as LLMs continue to evolve with increasing flexibility across successive iterations. While the BiLSTM architectures employed in this study, powered by GloVe embeddings, achieved commendable levels of accuracy, they nevertheless fell short of providing a foolproof solution. In what follows, three separate sections explore the technical observations regarding BiLSTM performance, the broader academic context around fairness and the hazards of automated penalties, and the limitations that shape the scope of the present research.

4.1 Technical Observations on Architecture and Embedding Strategies

The internal dynamics and performance outcomes of the BiLSTM models offer several instructive lessons regarding neural design and the pitfalls of pursuing endless complexity. The experiments that compared a "smaller" single-layer BiLSTM (256 units) to a deeper dual-layer variant revealed that a more compact architecture at times performed just as well—if not slightly better—than a network with additional stacked layers, at least when the problem is constrained to classifying synthetic vs. human-generated abstracts. This phenomenon emerged in both single-split experiments and K-fold cross-validation. One plausible explanation is that a single BiLSTM, when combined with well-selected hyperparameters and properly managed dropout, can capture the essential word transitions needed to differentiate AI-driven text from authentic human writing.

A recurring theme was the trade-off between trainable and static embeddings. Allowing GloVe vectors to be updated during backpropagation yielded slightly higher accuracy (peaking near 97%), along with a decrease in missed AI-generated samples (i.e., fewer false negatives). However, these refinements occasionally came at the cost of instability in validation curves, suggesting a tendency to overfit. Batch normalization, spatial dropout, and learning-rate scheduling alleviated this risk but did not remove it entirely. While many DL projects apply large or multiple LSTM layers in pursuit of higher capacity, these trials suggest that more complicated structures do not always produce proportionally better performance. In fact, the deeper, dual-layer BiLSTM risked over-parameterization, especially when embeddings themselves were already fine-tuned. Hence, the most pleasing balance in our scenario emerged when a single BiLSTM architecture combined adaptive word vectors with moderate dropout and an early-stopping policy.

The results also hint at the significance of domain suitability in pre-trained embeddings. Even though Word2Vec embeddings were larger in dimensionality (300 D) and originated from a broad corpus (Google

News), their coverage for specialized medical and academic terms was somewhat limited compared to GloVe's 100 D vectors. The coverage shortfall, about 81.66% for Word2Vec vs. 86.39% for GloVe, is arguably enough to tip classification performance in GloVe's favor, especially when detecting AI-generated abstracts in COVID-19-related research. In principle, an embedding corpus that closely mirrors the specialized domain—whether that is biomedical, epidemiological, or a different academic field—can make a substantial difference for classification tasks of this nature.

Risks of overfitting were mitigated through various strategies. Foremost, once the embedding layer was set to "trainable," the model had increased flexibility in shaping how it conceptualized textual input. This flexibility required more regularization. During ablation studies, authors discovered that removing spatial dropout entirely in certain static-embedding configurations boosted performance, since untrainable embeddings do not shift and thus may not over-memorize ephemeral patterns. Conversely, the models with adaptive embeddings typically benefited from having dropout inside the recurrent cells and in the dense layers, plus a stable learning-rate schedule. Early stopping on validation loss further kept overfitting in check, ensuring that the network did not spiral into memorizing idiosyncrasies in the training set.

An additional noteworthy observation is that smaller batch sizes, typically 32 or 64, facilitated more stable convergence in numerous model configurations. Such a finding is broadly consistent with research that suggests moderate batch sizes can strike a better balance between gradient noise and stable parameter updates [58]. This pattern was shown in multiple ablation runs, where going above a batch size of 128 sometimes produced suboptimal or vacillating validation results. On top of that, certain advanced techniques—like Hyperband optimization—proved beneficial for scanning broad hyperparameter spaces within the limited runtime environment that was used Li et al. [55].

Taken together, these experiments underscore the notion that, in certain cases, a simpler approach may be more effective. A single BiLSTM layer, combined with adaptive GloVe embeddings, moderate dropout, and a well-chosen learning rate, delivered performance that nearly matched or even surpassed more elaborate or deeper designs. That outcome is a salutary reminder: model complexity is no guarantee of superior classification, and smaller networks with fewer parameters often demonstrate more stable results, especially under practical constraints such as Kaggle's GPU resource limitations or when focusing on specialized tasks like detecting AI-generated abstracts.

4.2 Academic Perspective on Imperfect Detection and Potential Consequences

Throughout the evolution of LLMs—spanning GPT-3, GPT-4, DeepSeek, and their rapid successors, one consistent theme emerges. These models can produce text that can fool both automated detection tools and human reviewers as shown by Hakam et al. [10], Nabata et al. [12]. As progress in text-generation accelerates, the probability of achieving 100% classification success becomes even more remote. Despite achieving near 97% accuracy under optimal conditions, the experiments outlined here still resulted in the misclassification of a small subset of texts. That modest gap might seem acceptable to some, but in an academic context, even a small rate of false positives is enough to have considerable repercussions for students and researchers whose original work is unfairly labeled as machine-produced.

The evidence from this research, and referenced publications, indicates that detection systems are imperfect, and it is erroneous to treat a detection score as incontrovertible evidence of misconduct [6,13]. When institutions adopt a policy of penalties or zero-tolerance measures based purely on such tools, they create an environment where genuine originality may be questioned unfairly. Even the best DL solutions cannot account for all nuances of human creativity, disciplinary language, or specialized jargon. Some authors might use distinctive phrasings that overlap with AI-driven patterns, and these legitimate texts can be misread as synthetic.

It is also unrealistic to believe that a comprehensive detection system covering every domain, every new AI model, and every emergent writing style can be achieved. The rapidly evolving landscape—where new LLMs emerge on an almost weekly basis, continually fine-tuned on progressively diverse corpora—ensures that the classification challenge will remain a moving target. If a solution cannot deliver perfect or near-perfect reliability on a controlled set of GPT-3 abstracts, it becomes even less plausible to hope for total coverage across every discipline, model, and style. The risk of "hallucinations," where advanced text-generation systems produce plausible but inaccurate references or facts [28], only adds another layer of complexity in detection. Consequently, any strategy that treats automated outputs as the final word risks undue harm to students or authors.

In addition, fairness considerations suggest that educators should not penalize individuals solely based on a questionable detection score. A more balanced approach involves using these computational tools as preliminary filters, followed by thorough human review. In certain instances, the best safeguard is instructing students and researchers on responsible AI usage, clarifying how such systems may or may not be employed, and promoting learning and evaluation methods that require personal reflections, practical demonstrations, or contextual arguments that an automated text generator would struggle to replicate. Rather than placing our faith in a single numeric threshold from a detection model, academia might integrate such tool as part of a broader policy framework that includes training, transparent rules, and a measured process for further investigation. If, for instance, a piece of writing is flagged as "likely AI-generated," a department could provide the author with a chance to discuss or clarify how the text was produced before rendering any judgment.

Another important consideration revolves around fairness in grading and peer review. Students should not be penalized if a detection system incorrectly flags their authentic work. Likewise, if an advanced system evades detection by skillfully imitating academic style, educators should be prepared for the possibility that not everything that "looks real" is guaranteed to be genuine. Both educators and administrators thus face an ethical imperative to rely on multi-pronged methods that combine computational screening, domain expertise, and robust support for genuine academic inquiry. This viewpoint aligns with arguments made by Lawrence et al. [11], who caution that subjective impressions of textual quality do not always match authorship origin and can lead to misguided evaluations.

4.3 Limitations

Although the present study sheds considerable light on BiLSTM-based strategies for AI-text detection, several constraints limit the generalizability and broader applicability of the observed outcomes.

First, the investigation focused on GPT-3-generated abstracts from the AI-GA dataset [48], and these texts were paired with authentic COVID-19 abstracts. The domain specificity means that word usage and the rhetorical structures in the dataset may not represent other fields or more varied writing styles. In addition, GPT-3 is only one member of the LLM ecosystem, and the differences between GPT-4, DeepSeek, and other emergent models may pose new challenges. Performance might deviate if the system is tested on text from advanced, domain-tuned AI models or entirely different subject matter.

Second, although Hyperband optimization was employed to traverse a broad hyperparameter space, memory and runtime constraint in the Kaggle environment restricted the scope of exploration. More extensive searches or more specialized hyperparameters (e.g., focusing on even smaller or larger LSTM units, different decay schedules, or more exhaustive embedding sets) could potentially yield slightly higher accuracy or better generalization.

Third, the study's focus remained on performance metrics (accuracy, precision, recall, F1), without implementing methods to explain which textual cues the BiLSTM used to classify a text as AI-generated.

This limitation means that instructors or administrators might find it difficult to justify a detection result if asked to provide evidence beyond the model's numeric output. Additional techniques such as attention-weight visualization or layer-wise relevance propagation could clarify the signals that drive classification, offering a more transparent approach.

LLMs are updated continuously, and their capacity to produce more humanlike or domain-specific text expands over time. A classifier built around GPT-3 patterns risks gradual obsolescence if GPT-4 or emerging models introduce new phrasing habits, lexical patterns, or contextual depth that the original system did not anticipate. The immediate adoption of weekly or monthly model variations by malicious actors or even curious students implies that detection solutions require frequent re-validation and updating.

5 Conclusion

The comparison of single-layer and dual-layer BiLSTM architectures revealed that larger models do not always guarantee superior classification outcomes, particularly when the goal is to isolate AI-generated text among scientific abstracts. In many trials, a single bidirectional LSTM with 256 units, combined with pre-trained GloVe vectors, either met or surpassed the deeper network's performance. Although the trainable embedding option provided added precision and recall gains, it also introduced overfitting risks, thereby requiring careful tuning of dropout layers and learning rates. Even when best-case scenarios approached the 98.7% accuracy plateau achieved with previous LSTM-Word2Vec work [9], none of the models managed complete error-free detection.

These findings reaffirm that, regardless of their sophistication, automated classifiers cannot serve as the definitive authority in determining authorship. The danger of penalizing genuine text or overlooking meticulously crafted AI output underscores the value of a multi-layered approach, where computational methods serve as preliminary checkpoints rather than unilateral judges. Guidelines on responsible AI usage, along with assignments that invite reflections and unique insights, can further deter misuse and sustain true intellectual exploration. Rather than placing unwavering faith in a numeric verdict, one can integrate detection tools with well-planned academic policies, transparent review processes, and a clear path for authors to demonstrate the authenticity of their work. By adopting this more balanced perspective, it becomes possible to maintain a high standard of integrity and originality in a landscape where AI-driven text generation is advancing in sophistication every day.

Acknowledgement: The authors extend their gratitude to the creators and curators of the AI-GA dataset, whose effort made it possible to conduct this research in a rigorous and controlled manner. They also acknowledge the vibrant research community engaging with these complex questions of AI-generated text, from whom they have drawn inspiration and methodological insights. Finally, they are thankful for the computational platform support that allows experimentation under realistic constraints.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors are responsible for all aspects of this research, including conceptualization, methodology design, data analysis, model development, experimentation, and manuscript preparation. The authors confirm contribution to the paper as follows: study conception and design: Lilia-Eliana Popescu-Apreutesei, Mihai-Sorin Iosupescu, Sabina Cristiana Necula and Vasile-Daniel Păvăloaia; analysis and interpretation of results: Lilia-Eliana Popescu-Apreutesei and Mihai-Sorin Iosupescu; draft manuscript preparation: Lilia-Eliana Popescu-Apreutesei, Mihai-Sorin Iosupescu, Sabina Cristiana Necula and Vasile-Daniel Păvăloaia. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in GitHub at https://github.com/panagiotisanagnostou/AI-GA (accessed on 22 February 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.
2. OpenAI. ChatGPT: optimizing language models for dialogue [Internet]. [cited 2025 Feb 20]. Available from: https://openai.com/blog/chatgpt/.
3. Bi X, Chen D, Chen G, Chen S, Dai D, Deng C, et al. Deepseek LLM: scaling open-source language models with longtermism. arXiv:2401.02954. 2024.
4. Zhu Q, Guo D, Shao Z, Yang D, Wang P, Xu R, et al. DeepSeek-Coder-V2: breaking the barrier of closed-source models in code intelligence. arXiv: 2406.11931. 2024.
5. Jawahar G, Sagot B, Seddah D. What does BERT learn about the structure of language? In: Proceedings of the Annual Meeting of the Association for Computational Linguistics; 2019 Jul 28–Aug 2; Florence, Italy.
6. Weber-Wulff D, Anohina-Naumeca A, Bjelobaba S, Folttfytfnek T, Guerrero-Dib J, Popoola O, et al. Testing of detection tools for AI-generated text. Int J Educ Integr. 2023;19(1):26. doi:10.1007/s40979-023-00146-z.
7. Cotton D, Cotton P, Shipway J. Chatting and cheating: ensuring academic integrity in the era of ChatGPT. Innov Educ Teach Int. 2024;61(2):228–39. doi:10.1080/14703297.2023.2190148.
8. Bhullar P, Joshi M, Chugh R. ChatGPT in higher education—a synthesis of the literature and a future research agenda. Educ Inf Technol. 2024;29(16):21501–22. doi:10.1007/s10639-024-12723-x.
9. Theocharopoulos PC, Anagnostou P, Tsoukala A, Georgakopoulos SV, Tasoulis SK, Plagianakos VP. Detection of fake generated scientific abstracts. In: Proceedings of the 2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService); 2023 Jul 17–20; Athens, Greece.
10. Hakam HT, Prill R, Korte L, Lovreković B, Ostojić M, Ramadanov N, et al. Human-written vs AI-generated texts in orthopedic academic literature: comparative qualitative analysis. JMIR Form Res. 2024;8:e52164.
11. Lawrence KW, Habibi AA, Ward SA, Lajam CM, Schwarzkopf R, Rozell JC. Human versus artificial intelligence-generated arthroplasty literature: a single-blinded analysis of perceived communication, quality, and authorship source. Int J Med Robot Comput Assist Surg. 2024;20(1):e2621. doi:10.1002/rcs.2621.
12. Nabata KJ, AlShehri Y, Mashat A, Wiseman SM. Evaluating human ability to distinguish between ChatGPT-generated and original scientific abstracts. Updates Surg. 2025;1-7(3):e105. doi:10.1007/s13304-025-02106-3.
13. Kim HJ, Yang JH, Chang DG, Lenke LG, Pizones J, Castelein R, et al. Assessing the reproducibility of the structured abstracts generated by ChatGPT and Bard compared to human-written abstracts in the field of spine surgery: comparative analysis. J Med Internet Res. 2024;26:e52001.
14. Shcherbiak A, Habibnia H, Böhm R, Fiedler S. Evaluating science: a comparison of human and AI reviewers. Judgm Decis Mak. 2024;19:e21. doi:10.1017/jdm.2024.24.
15. Cheng SL, Tsai SJ, Bai YM, Ko CH, Hsu CW, Yang FC, et al. Comparisons of quality, correctness, and similarity between ChatGPT-generated and human-written abstracts for basic research: cross-sectional study. J Med Internet Res. 2023;25(1):e51229.
16. Chowdhury SA, Almerekhi H, Kutlu M, Keleş KE, Ahmad F, Mohiuddin T, et al. AI vs. human—academic essay authenticity challenge. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
17. Jiao K, Yao X, Ma S, Fang S, Guo Z, Xu B, et al. Leveraging multilingual proxy LLMs for machine-generated text detection in academic essays. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.

18. Gharib R, Elgendy A. Fine-tuned language models for detection of academic authenticity, results and thoughts. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
19. Agrahari S, Jayant S, Kumar S, Singh SR. Guardians of academic integrity: multilingual detection of AI-generated essays. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
20. Indurthi V, Varma V. Fast and scalable method for detection of academic essay authenticity. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
21. Varadarajan V, Giorgi S, Mangalik S, Soni N, Markowitz DM, Schwartz HA. The consistent lack of variance of psychological factors expressed by LLMs and spambots. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
22. Creo A, Pudasaini S. SilverSpeak: evading AI-generated text detectors using homoglyphs. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
23. Khlaif ZN, Mousa A, Hattab MK, Itmazi J, Hassan AA, Sanmugam M, et al. The potential and concerns of using AI in scientific research: chatGPT performance evaluation. JMIR Med Educ. 2023;9:e47049.
24. Kumar A, Kumar A, Bhoyar S, Mishra AK. Does ChatGPT foster academic misconduct in the future? Public Adm Policy. 2024;27(2):140–53. doi:10.1108/pap-05-2023-0061.
25. Zhang L, Amos C, Pentina I. Interplay of rationality and morality in using ChatGPT for academic misconduct. Behav Inf Technol. 2025;44(3):491–507. doi:10.1080/0144929x.2024.2325023.
26. Wang J, Cornely PR. Addressing academic misconduct in the age of ChatGPT: strategies and solutions. In: Proceedings of the 2023 7th International Conference on Education and E-Learning; 2023 Nov 25–27; Tokyo, Japan.
27. Májovský M, Černý M, Kasal M, Komarc M, Netuka D. Artificial intelligence can generate fraudulent but authentic-looking scientific medical articles: pandora's box has been opened. J Med Internet Res. 2023;25:e46924.
28. Hua HU, Kaakour AH, Rachitskaya A, Srivastava S, Sharma S, Mammo DA. Evaluation and comparison of ophthalmic scientific abstracts and references by current artificial intelligence chatbots. JAMA Ophthalmol. 2023;141(9):819–24. doi:10.1001/jamaophthalmol.2023.3119.
29. Wang Z, Zhou C. Reassessing AI in medicine: exploring the capabilities of AI in academic abstract synthesis. J Med Internet Res. 2024;26(1):e55920.
30. Onan A, Çelikten T. Evaluating the coherence and diversity in AI-generated and paraphrased scientific abstracts: a fuzzy topic modeling approach. In: Proceedings of the International Conference on Intelligent and Fuzzy Systems; 2024 Jul 16–18; Canakkale, Türkiye.
31. Maktabdar Oghaz M, Babu Saheer L, Dhame K, Singaram G. Detection and classification of ChatGPT generated contents using deep transformer models. Front Artif Intell. 2025;8:1458707. doi:10.3389/frai.2025.1458707.
32. Yadagiri A, Lekkala ST, Vardhan MS, Pakray P, Krishna RM. AI-generated text using transformer-based approaches. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
33. Mobin MK, Islam MS. LuxVeri at GenAI detection task 3: cross-domain detection of AI-generated text using inverse perplexity-weighted ensemble of fine-tuned transformer models. arXiv:2501.11918v1. 2025.
34. Marchitan TG, Creanga C, Dinu LP. Team Unibuc—NLP at GenAI detection task 1: Qwen it detect machine-generated text?. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
35. Abiola TO, Bizuneh TA, Uroosa F, Hafeez N, Sidorov G, Kolesnikova O, et al. Advancing multilingual machine-generated text detection. In: Proceedings of the 31st International Conference on Computational Linguistics; 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
36. Mohamed TA, Khafgy MH, ElSedawy AB, Ismail AS. A proposed model for distinguishing between human-based and ChatGPT content in scientific articles. IEEE Access. 2024;12:121251–60. doi:10.1109/access.2024.3448315.

37. AL-Smadi M. Detecting machine-generated academic essays in English and Arabic using ELECTRA and stylometry. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
38. Zhang Z, Chen S, Liu B. Enhancing machine-generated text detection with semantic and probabilistic features. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
39. Emi B, Spero M, Masrour E. Pangram at GenAI detection task 3: an active learning approach to machine-generated text detection. In: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect); 2025 Jan 19–24; Abu Dhabi, United Arab Emirates.
40. Mese I. Tracing the footprints of AI in radiology literature: a detailed analysis of journal abstracts. RöFo-Fortschritte Auf Dem Geb Der Röntgenstrahlen Und Der Bildgeb Verfahr. 2024;196(8):843–9. doi:10.1055/a-2224-9230.
41. Makiev KG, Asimakidou M, Vasios IS, Keskinis A, Petkidis G, Tilkeridis K, et al. A study on distinguishing ChatGPT-generated and human-written orthopaedic abstracts by reviewers: decoding the discrepancies. Cureus. 2023;15(11):e49166. doi:10.7759/cureus.49166.
42. Alencar-Palha C, Ocampo T, Silva TP, Neves FS, Oliveira ML. Performance of a generative pre-trained transformer in generating scientific abstracts in dentistry: a comparative observational study. Eur J Dent Educ. 2025;29(1):149–54. doi:10.1111/eje.13057.
43. Carnino JM, Chong NYK, Bayly H, Salvati LR, Tiwana HS, Levi JR. AI-generated text in otolaryngology publications: a comparative analysis before and after the release of ChatGPT. Eur Arch Otorhinolaryngol. 2024;281(11):6141–6. doi:10.1007/s00405-024-08834-3.
44. Howard FM, Li A, Riffon MF, Garrett-Mayer E, Pearson AT. Characterizing the increase in artificial intelligence content detection in oncology scientific abstracts from 2021 to 2023. JCO Clin Cancer Inform. 2024;8(8):e2400077. doi:10.1200/CCI.24.00077.
45. Gralha JG, Pimentel AS. Gotcha GPT: ensuring the integrity in academic writing. J Chem Inf Model. 2024;64(21):8091–7. doi:10.1021/acs.jcim.4c01203.
46. Hamed AA, Wu X. Detection of ChatGPT fake science with the xFakeSci learning algorithm. Sci Rep. 2024;14(1):16231. doi:10.1038/s41598-024-66784-6.
47. Gao C, Howard F, Markov N, Dyer E, Ramesh S, Luo Y, et al. Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. BioRxiv. 2022;12(12):521610. doi:10.1101/2022.12.23.521610.
48. Anagnostou P. Panagiotisanagnostou/AI-GA [Internet]. [cited 2025 Apr 29]. Available from: https://github.com/panagiotisanagnostou/AI-GA.
49. Bird S, Klein E, Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. Sebastopol, CA, USA: O'Reilly Media; 2009. 504 p.
50. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. doi:10.3115/v1/d14-1162.
51. Mikolov T, Chen K, Corrado GS, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations; 2013 May 2–4; Scottsdale, AZ, USA.
52. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;45(11):2673–81. doi:10.1109/78.650093.
53. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
54. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proc Mach Learn Res. 2015;37:448–56.
55. Li L, Jamieson K, Rostamizadeh A, Gonina E, Hardt M, Recht B, et al. A system for massively parallel hyperparameter tuning. Proc Mach Learn Syst. 2020;2:230–46.

56. Meyes R, Lu M, Waubert de Puiseau C, Meisen T. Ablation studies in artificial neural networks. arXiv:1901.08644v2. 2019.
57. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Vol. 1. Cambridge, MA, USA: MIT Press; 2016. 800 p.
58. Kingma D, Ba J. Adam: a method for stochastic optimization. arXiv:1412.6980. 2014.
59. Prechelt L. Automatic early stopping using cross validation: quantifying the criteria. Neural Netw. 1998;11(4):761–7. doi:10.1016/s0893-6080(98)00010-0.
60. Smith LN. Cyclical learning rates for training neural networks. In: Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV); 2017 Mar 24–31; Santa Rosa, CA, USA. doi:10.1109/WACV.2017.58.
61. Cox DR. The regression analysis of binary sequences. J R Stat Soc Ser B Stat Methodol. 1958;20(2):215–32. doi:10.1111/j.2517-6161.1958.tb00292.x.
62. Lewis DD. Naive (Bayes) at forty: the independence assumption in information retrieval. In: Proceedings of the Machine Learning: ECML-98; 1998 Apr 21–23; Chemnitz, Germany. doi:10.1007/bfb0026666.
63. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. doi:10.1023/A:1010933404324.
64. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Statist. 2001;29(5):1189–232. doi:10.1214/aos/1013203451.
65. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13–17; San Francisco, CA, USA. doi:10.1145/2939672.2939785.
66. Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. doi:10.3115/v1/d14-1181.
67. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000;12(10):2451–71. doi:10.1162/089976600300015015.
68. McNemar Q. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika. 1947;12(2):153–7. doi:10.1007/BF02295996.
69. Student. The probable error of a mean. Biometrika. 1908;6(1):1–25. doi:10.2307/2331554.
70. Wilcox RR. Modern statistics for the social and behavioral sciences: a practical introduction. Boca Raton, FL, USA: CRC Press; 2011. 862 p.
71. Efron B, Tibshirani RJ. An introduction to the bootstrap. Boca Raton, FL, USA: CRC Press; 1994. 456 p.
72. Ioffe S. Batch renormalization: towards reducing minibatch dependence in batch-normalized models. arXiv:1702.03275v2. 2017.