Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051574

ARTICLE

HybridGAD: Identification of AI-Generated Radiology Abstracts Based on a Novel Hybrid Model with Attention Mechanism

Tuğba Çelikten[1] **and Aytuğ Onan**[2,*]

[1]Department of Software Engineering, Faculty of Technology, Manisa Celal Bayar University, Manisa, 45140, Turkey

[2]Department of Computer Engineering, Faculty of Engineering and Architecture, İzmir Katip Çelebi University, İzmir, 35620, Turkey

*Corresponding Author: Aytuğ Onan. Email: aytugonan@gmail.com

Received: 09 March 2024 Accepted: 24 June 2024 Published: 15 August 2024

ABSTRACT

The purpose of this study is to develop a reliable method for distinguishing between AI-generated, paraphrased, and human-written texts, which is crucial for maintaining the integrity of research and ensuring accurate information flow in critical fields such as healthcare. To achieve this, we propose HybridGAD, a novel hybrid model that combines Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Bidirectional Gated Recurrent Unit (Bi-GRU) architectures with an attention mechanism. Our methodology involves training this hybrid model on a dataset of radiology abstracts, encompassing texts generated by AI, paraphrased by AI, and written by humans. The major findings of our analysis indicate that HybridGAD achieves a high accuracy of 98%, significantly outperforming existing state-of-the-art models. This high performance is attributed to the model's ability to effectively capture the contextual nuances and structural differences between AI-generated and human-written texts. In conclusion, HybridGAD not only enhances the accuracy of text classification in the field of radiology but also paves the way for more advanced medical diagnostic processes by ensuring the authenticity of textual information. Future research will focus on integrating textual and visual data for comprehensive radiology assessments and improving model generalization with partially labeled data. This study underscores the potential of HybridGAD in transforming medical text classification and highlights its applicability in ensuring the integrity and reliability of research in healthcare and beyond.

KEYWORDS

Generative artificial intelligence; AI-generated text detection; attention mechanism; hybrid model for text classification

1 Introduction

Text classification, one of the subfields of natural language processing (NLP), is the process of automatically assigning a specific textual expression to one or more predefined classes. Text classification is commonly employed in various artificial intelligence (AI) applications, such as sentiment analysis (to gauge emotional tone), spam filtering (to distinguish spam from genuine content), topic labelling (to categorize textual subjects), financial analysis (to identify specific financial situations), and social media analysis (to classify user comments and posts) [1,2]. Moreover, in the field

of medicine, text classification is frequently utilized for information extraction purposes in patient reports, diagnosis, and medical analysis [3]. This AI technique plays a crucial role in categorizing and understanding textual data in different domains, showcasing its versatility in applications across multiple industries.

Recently, there has been great interest in text-generation tools that can generate human-like text such as ChatGPT, developed by OpenAI. ChatGPT is used for varying tasks for example text generation, question-answering, text translation, etc. Even if these tools show excellent improvement in the generation of human-like text, differences between human-written and AI-generated text should be detected [4]. Equally important, the identification of AI-generated scientific abstracts from human-written scientific text is crucial for maintaining the integrity and reliability of academic research. As AI continues to advance, so does its capability to generate content that closely mimics human writing. This poses a significant challenge in ensuring that scientific literature remains authentic and trustworthy. By differentiating AI-generated abstracts from human-written, researchers and publishers maintain academic integrity, prevent misinformation, and uphold scientific discourse credibility. Moreover, this distinction ensures transparency and accountability in the scientific community. It enables researchers to evaluate information origins and validity, fostering advancements grounded in genuine scientific inquiry.

The text classification tasks taken hand in this context, traditional text classification tasks are carried out based on fundamental machine learning (ML) algorithms, following the steps of data collection, data preprocessing, and model implementation. During the data preprocessing stage, informative attributes or features are identified to create a distinction, and then these features are used to train a ML algorithm model for prediction. It is crucial that the features considered at this stage preserve the context expressed in the text. In this regard, deep learning (DL) algorithms and transformer models have proven to achieve highly performant results in text classification tasks.

DL employs a multi-layered approach to the hidden layers of a neural network. The DL approach, a type of complex network consisting of many interconnected neurons, has been widely used in a wide range of applications due to the excellent performance of artificial neural networks (ANNs) in copying and reasoning with the human brain. Traditional ML algorithms manually define and extract features or use feature selection methods. On the other hand, DL models automatically train and extract features, resulting in improved accuracy and performance. DL is highly effective in overcoming the challenges of NLP and, consequently, in text classification, as it can comprehend complex connections between words and expressions. Especially Long Short-Term Memory (LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU) are deep neural network algorithms that have high success in this field.

LSTM neural networks perform better in processing data types such as time series and textual expressions because, the "memory cells" in LSTM structure, can effectively capture long-term dependencies between sequences of words. Running in this way, LSTM models can maintain long-term dependencies of information.

Bidirectional Long Short-Term Memory (Bi-LSTM) is a version of LSTM architecture that consists of two independent LSTM networks. Bi-LSTM operates by sequentially capturing information in both forward and backward directions during input processing such as it analyses a sentence from the beginning to the end (left-to-right) and then from the end to the beginning (right-to-left).

A neural network model that operates similarly to LSTM is the GRU, an enhanced version of LSTM. The typical GRU architecture replaces LSTM's memory cell with a single update gate, simplifying the memory structure while introducing two gates known as the Reset Gate and Update

Gate. GRU is a simplified version of LSTM, featuring fewer parameters. Consequently, it is often preferred for faster training processes and when dealing with limited data situations.

To utilize these models, it is necessary to transform the dataset into numerical representations. For this purpose, during the preprocessing stage, tokenizer techniques are employed to convert sentences into tokens and numerical vector representations. In this process, raw text is initially segmented into words or sub words, and these are subsequently transformed into unique integer identifiers through a lookup table.

In our study, we developed a hybrid model named 'HybridGAD' by combining the advantages of LSTM, Bi-LSTM, and GRU models due to their specialized architectures in understanding context in text processing. The aim was to create a hybrid architecture to gain a more comprehensive understanding by leveraging the strengths of these models in handling complexities in textual data. The main contributions of the proposed architecture can be summarized as follows:

- LSTM addresses long-term dependencies and regulates information via memory cells and gates, while Bi-LSTM integrates information bidirectionally from both directions during input sequence analysis. Additionally, GRU offers a simplified structure, fewer parameters, and fast training characteristics, contributing to a more robust architecture.
- We chose the Bio_ClinicalBERT [5] tokenizer over traditional methods for processing radiology article abstracts, as it specializes in handling medical terminology. Bio_ClinicalBERT is specifically trained to handle medical complexities in textual data. In this context, we aimed to achieve performance beyond standards by ensuring that our proposed model better understands medical terminology.
- To enable the model to learn important details in medical texts more precisely and perform the classification task more accurately, we utilized the attention mechanism. The attention mechanism, adopted in DL models, shares similar characteristics in conveying where and what needs to be emphasized. It allows models to focus on crucial parts of input data, especially in long texts or large datasets, by ignoring irrelevant information. Additionally, the attention mechanism contributes to the model's better understanding of context and extraction of more effective features. In this way, we aimed for the proposed model to deliver more robust and consistent results.

The HybridGAD model has the potential to provide a solid foundation for the identification of radiology-focused AI-generated texts. Efforts to integrate this model into practical scenarios in clinical settings can significantly increase its applicability, thereby improving the efficiency of medical diagnostic procedures. Such efforts may serve as catalysts for future research efforts aimed at advancing the development of radiology text classification. The HybridGAD model is capable of distinguishing between human-written texts and AI-generated texts. By examining both grammatical structure and content features, this model can detect whether the text was generated by humans or AI. Besides, the proposed model is also capable of identification of GPT-paraphrased versions of texts written by humans with different words. These capabilities are improved by hybrid DL algorithms and extensive data. So, the model can better analyse the grammatical structure and content features of texts, while the large dataset helps the model learn different writing styles and language usage. Additionally, the model's processing of texts with specialized tools such as the Bio_ClinicalBERT tokenizer helps it more precisely identify specific terms and phrases in healthcare texts.

As a result, the HybridGAD model stands out for its ability to distinguish differences between texts written by humans and texts generated by GPT. This helps make decisions based on reliable and accurate information, especially in the field of healthcare, while also contributing to the detection of manipulative or misleading content.

2 Related Works

In this section, the previous research related to the study field have been examined under two main headings. Firstly, general models, detect generated text by AI using ML and DL algorithms, etc., are investigated. Secondly, models utilizing hybrid approaches are examined.

2.1 General Models for Detection AI Generated Text

The development of methods to detect AI-generated text has been an area of active research, with several studies making substantial contributions. Verma et al. [6] introduced the "Ghostbuster" system, a novel approach that leverages a series of weaker language models to identify AI-generated content. By conducting a structured search over possible combinations of features and training a classifier based on selected features, their system was able to achieve a remarkable F1-score of over 98.4 across diverse datasets including news, student essays, and creative writing. This success highlights the system's capability in detecting AI-generated text across various domains and prompts, setting a high benchmark in the field.

Building on the concept of generalization in detection, Mitchell et al. [7] proposed "DetectGPT," a zero-shot detection method that assesses the probability function's negative curvature to determine if a text is machine-generated. Their method shows enhanced accuracy in specific contexts like XSum stories and SQuAD Wikipedia articles and demonstrates superior generalization capabilities compared to traditional supervised detectors, especially in new languages and domains.

Similarly, Cingillioglu [8] focused on distinguishing texts generated by OpenAI's ChatGPT using a bespoke language model trained on a balanced dataset of human and AI-generated articles. By exploiting n-gram differences and employing a Support Vector Machine (SVM) classifier, the study successfully differentiated between human and AI-generated texts, achieving high accuracy.

Ibrahim [9] addressed a related but slightly different problem: detecting AI-assisted plagiarism. Using a dataset comprised of texts from humans and ChatGPT, Karim developed classifiers based on a Robustly Optimized BERT Pretraining Approach (RoBERTa) based approaches and the GPT-2 Output Detector Demo. While generally successful, the performance of these classifiers varied, indicating the influence of dataset characteristics on detection efficacy.

Lastly, Lee et al. [10] explored the linguistic features of fake reviews generated by AI, employing a variety of machine learning techniques including Logistic Regression, Random Forest, SVM, Neural Networks, and AdaBoost on a large dataset of Kaggle reviews. Their findings reveal challenges in differentiating between human and machine-generated reviews, suggesting the need for more sophisticated models and a deeper analysis of language model training data.

Each of these studies informs our current research by illustrating different approaches and outcomes in the detection of AI-generated text. Our work builds on this foundation with a hybrid model that integrates LSTM, Bi-LSTM, and Bi-GRU architectures, aiming to enhance detection accuracy and robustness, particularly in handling AI-paraphrased texts—a gap not specifically addressed by previous studies. This approach not only broadens the scope of detectable manipulations in text but also contributes to the ongoing dialogue on improving AI detection methodologies in scholarly and practical applications.

2.2 Hybrid Models for Detection AI Generated Text

Recent advancements in AI text generation necessitate sophisticated methods to detect such content accurately. Studies in this domain have increasingly focused on distinguishing between

human-generated and AI-generated texts, employing a variety of DL models and techniques. Sadiq et al. [11] developed a methodology using a dataset of 25,572 tweets, where they applied feature extraction techniques such as Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF), alongside FastText embeddings. They demonstrated that a hybrid model combining Convolutional Neural Network (CNN) and LSTM networks outperformed standalone models, showcasing the efficacy of hybrid approaches in text classification. Similarly, Oketunji [12] explored hybrid DL models, integrating CNN and Recurrent Neural Network (RNN) architectures to differentiate AI-generated texts from human-written ones. This study underscored the potential of hybrid models to achieve high accuracy across various AI models like GPT-3.5 and GPT-4. Furthering the discussion on language models, Gaggar et al. [13] utilized RoBERTa-based models trained on datasets from diverse sources to identify differences between texts generated by LLMs and humans. Their findings highlighted the critical role of sentence length and the robust performance of RoBERTa models in detecting nuanced differences. Gambetti et al. [14] introduced a multimodal model, FLAVA, which integrated text and image data to detect fake content. This approach was found to significantly outperform unimodal models, indicating the added value of leveraging multiple data types in deepfake detection. In a focused study on Arabic texts, Harrag et al. [15] developed the AraBERT model, which incorporated an attention mechanism to distinguish between human and AI-generated tweets. This model demonstrated superior performance over various RNN-based models, illustrating the benefits of attention mechanisms in enhancing model accuracy.

Katib et al. [16] employed the Tunicate Swarm Algorithm with LSTM for optimizing feature extraction processes, achieving high accuracy in differentiating between human and AI-generated texts. This approach emphasized the importance of parameter optimization in improving classification tasks.

Finally, Gambini et al. [17] introduced TriFuseNet, a novel network that combines stylistic features, language model embeddings, and character-based features for deepfake text detection. Despite its complex architecture, it provided valuable insights into the capabilities of multi-branch networks in text classification.

The state-of-the-art models have been summarized in the Table 1. Building on these foundational studies, our research introduces a hybrid model that not only leverages LSTM, Bi-LSTM, and Bi-GRU architectures but also incorporates a domain-specific tokenizer, Bio_ClinicalBERT, for enhanced performance in detecting paraphrased AI-generated texts. Our model aims to address the gaps in effectively parsing complex AI-generated and paraphrased texts by integrating the strengths of both general and hybrid modelling approaches. Through comprehensive experiments and evaluations, we strive to validate the effectiveness and reliability of our model, contributing to the broader discourse on AI text detection.

Table 1: Summary of studies on detecting AI-generated text

Author	Gen. AI method	Details for the architecture	Attention mechanism	Proposed model name
Verma et al. [6]	GPT-3.5-Turbo	RoBERTa	Not implemented	GHOSTBUSTER
Mitchell et al. [7]	GPT-NeoX	Zero-shot detectors, GPT2_TOKENIZER	Not implemented	DetectGPT

(Continued)

Table 1 (continued)

Author	Gen. AI method	Details for the architecture	Attention mechanism	Proposed model name
Cingillioglu [8]	ChatGPT	BOW + SVM	Not implemented	n-gram discrepancy BOW
Ibrahim [9]	ChatGPT	RoBERTa	Not implemented	GPT-2 output detector demo, crossplag detector
Lee et al. [10]	GPT-2	Logistic regression, random forest, SVM, neural networks, AdaBoost	Not implemented	—
Sadiq et al. [11]	Markov chains, RNN, GPT-2, GPT-3	TF, TF-IDF, FastText and FastText sub word, CNN, LSTM and CNN-LSTM	Not implemented	—
Oketunji [12]	GPT-3.5, GPT-4, PaLM2, LLaMa2	Hybrid layer output = CNN + RNN	Not implemented	—
Gaggar et al. [13]	GPT-3.5-Turbo API	RoBERTa-base and RoBERTa-large	Not implemented	—
Gambetti et al. [14]	GPT-4-Turbo, DALL-E-2	GPTNeo, ResNet-50, RoBERTa, BERT, GPT-3, CLIP	Not implemented	AiGen-FoodReview, dataset, FLAVA
Harrag et al. [15]	GPT2-Small-Arabic	LSTM, Bi-LSTM, GRU, BI-GRU	Bidirectional attention network	AraBERT
Katib et al. [16]	ChatGPT	TF-IDF, CountVectorizer, TSA, LSTM	Not implemented	TSA-LSTMRNN
Gambini et al. [17]	text-davinci-003	Stylistic features, contextualized sentence embeddings and Char-CNN	Not implemented	Fine-tuned BERTweet, TriFuseNet

3 Research Gaps

Nowadays, with the improvements in data science, progress has been made in text classification. In this way many classification tasks that were done manually have been automated. Thanks to the studies mentioned in Section 2, these benefits of artificial intelligence are used in a wide range of areas, from education to health. In addition, with the development of generative AI tools, text classification has shifted to a different field. Texts generated by generative AI tools such as ChatGPT are very similar to those generated by humans. For this reason, they have become the most used aids for people with their advanced ability in a wide range of subjects, including writing articles, generating code for programming, marketing, and health [18]. The main reason why people use these tools is that such texts are very difficult to distinguish from the human eye. This situation leads to problems such as

plagiarism cases, the increase of inauthentic academic studies, and the spread of unreal information. Especially in the field of medicine, patients can be misled by accessing information that appears to be written by doctors. To find a solution to these problems, there are many studies in the literature that distinguish between AI-generated and human-written texts. Studies have been carried out, especially on models developed with deep learning methods. The study we are considering is a hybrid model created with deep learning models. Our goal is to build a more durable model by combining the power of deep learning methods. Models such as LSTM and Bi-GRU, which are the most preferred in the field of text processing, perform text analysis more accurately when used together.

Our model is a domain-specific model, meaning it is trained with a dataset consisting of real data specific to the field of radiology. This makes our study superior to existing studies. In addition, since it is a field-based study, the tokenization technique used in the embedding phase was chosen accordingly. This is the purpose of using Bio_ClinicalBERT as a tokenizer. With Bio_ClinicalBERT, texts are analysed by preserving their context, and weighting is done accordingly. Therefore, the meanings of the terms used in the field of radiology are preserved on a subject-by-subject basis.

4 Proposed Architecture

Our proposed architecture consists of a various layer which are data pre-processing, an input layer, an LSTM layer, a bi-directional LSTM layer, a layer containing an attention mechanism, bi-directional GRU layer and pooling layer, a fully connected layer and output layers, as shown in Fig. 1. During preprocessing, we apply data preprocessing techniques for spelling correction and noise removal. Using a word representation technique, we vectorize the representations of words, in the input layer. We extract features and obtain the long dependencies feature with LSTM architecture. Bi-LSTM takes the LSTM-generated features as input and obtains the long dependencies feature to capture the context during training. The attention mechanism added after Bi-LSTM to allow focusing on important words. The obtained dependencies and weighted features are given as input to the Bi-GRU layer. Each layer is discussed in detail in the following sections.

4.1 Data Pre-Processor

Preprocessing consists of several steps performed to prepare the text for model. In NLP tasks, the raw material of the text analysis process is words [19]. So, the word representations as an input for the model should not cause errors or loss of information. For this purpose, some processes are applied to clean the data set and make it clean before classification. In the study, in the pre-processing step; removal of special characters, detection of URLs, user-mention expressions, emojis, removal spaces, lowercase letter conversion, and removal of punctuation have been implemented. We also made a statistical analysis of dataset during the pre-processing stage and showed it in Table 2. After the preprocessing process, the clean version of the data was shown in Table 2.

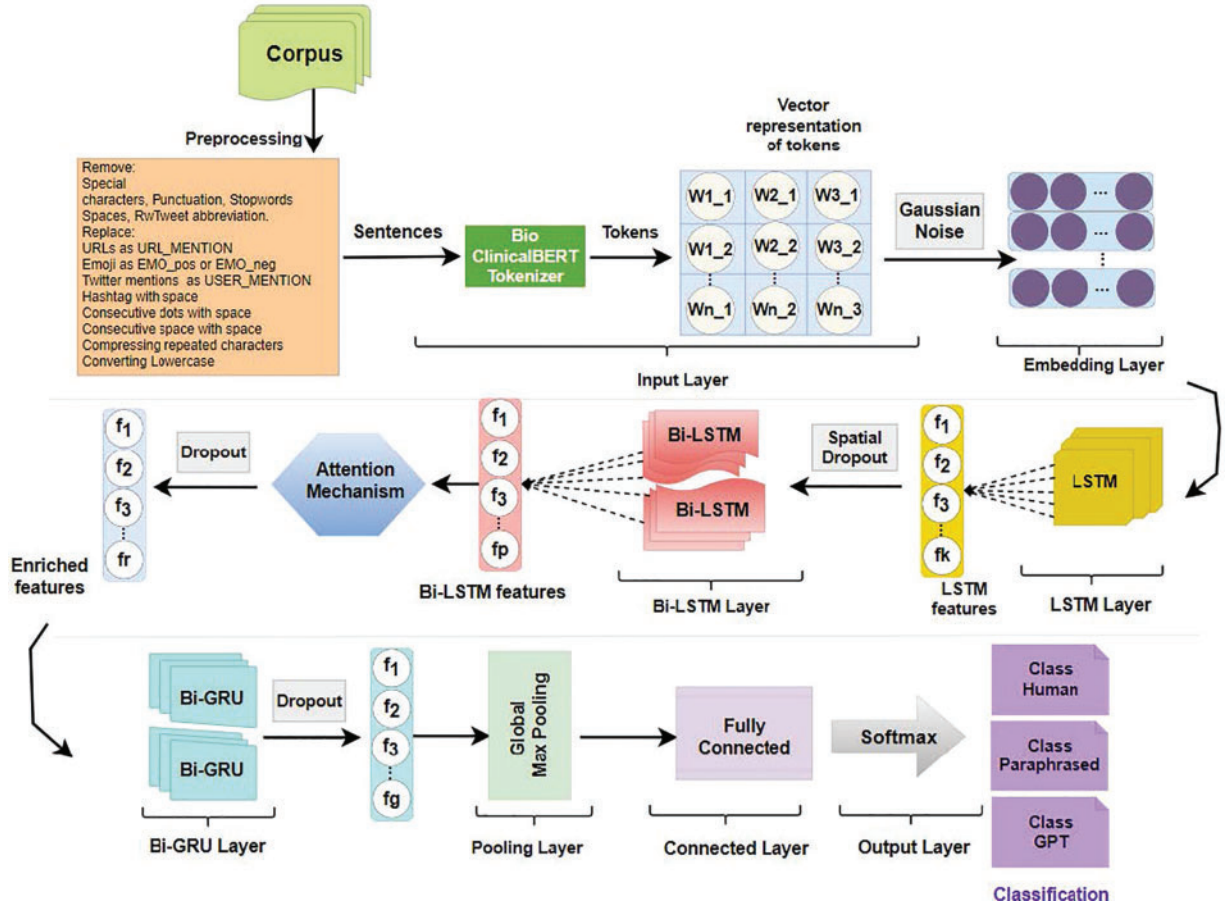

Figure 1: Proposed model architecture

Table 2: Examples of original and pre-processed data

The original data	After pre-processing
Based on the workflows of radiographers and radiologists the records were organized by body part (http://www.idr.med.uni-erlangen.de/orthorad/orthorad.htm, accessed on 01 Apr. 2024).	Based on the workflows of radiographers and radiologists the records were organized by body part URL.
A dedicated radiology education Twitter account (@ctisus) based at Johns Hopkins Hospital. Additionally, the study used.	A dedicated radiology education twitter account USER_MENTION based at johns hopkins hospital additionally the study used.

4.2 Input Layer

First of all, it is required to create representations of textual expressions. There are several domain-specific text representation methods developed for specific fields. Examples of these methods include Bio_ClinicalBERT designed for use in medical research, PubMedBERT [20] tailored for processing biomedical literature, EduBERT [21] for the education domain, and Stock2Vec for the finance domain. Since our dataset includes radiology article abstracts, we utilized the Bio_ClinicalBERT model to conduct a context-focused investigation, aiming to enhance the classification performance.

Since our dataset includes radiology article abstracts, we used the Bio_ClinicalBERT model, which was developed by fine-tuning the BioBERT [22] model with the data in the MIMIC-III database. In this way, we applied Bio_ClinicalBERT to increase the classification performance and robustness of the model by conducting a context-oriented study. The Bio_ClinicalBERT model was trained on the entirety of notes from MIMIC III, a database encompassing electronic health records of ICU patients at Beth Israel Hospital in Boston, MA, USA. Using Bio_ClinicalBERT, each sentence in the dataset is divided into tokens. Special tokens [CLS] (start) and [SEP] (end) are added to each sentence, indicating the beginning and end of the input.

Each token is converted into a numerical representation assigned by the Bio_ClinicalBERT. As a result, the data is formatted at each step into a representation that the model can understand, and then the model is trained on this numerical representation. The tokenization process using Bio_ClinicalBERT is shown in Fig. 2.

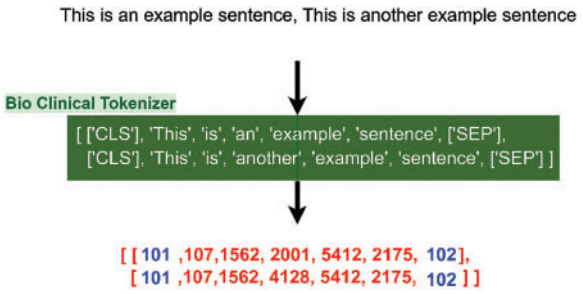

Figure 2: Tokenization process of Bio_ClinicalBERT

4.3 Long Short-Term Memory

LSTM is a RNN method commonly used in processing sequential data types such as time series or text. In text processing, where preserving long-term dependencies, or context, is a primary concern, LSTM is often the preferred choice. Due to its inherent structure, LSTM has the ability to retain information from previous time steps, allowing it to use this information in making predictions for the future. In NLP, each word in a sentence is sequentially connected to one another, and the meaning of words can vary based on the context of preceding and succeeding words. Therefore, LSTM is frequently chosen in text analysis tasks in order to capture and utilize these sequential dependencies effectively.

LSTM architecture of a recurrent module is illustrated in Fig. 3. Memory cells with self-connections are utilized to store the temporal state of the network within memory blocks. To control the information flow, LSTM employs unique multiplicative units called gates. Each memory block in the LSTM architecture has an input gate and an output gate. The input gate is responsible for allowing the flow of input activations into the memory cell. On the other hand, the output gate

is responsible for allowing the flow of cell activations to the rest of the network. The forget gate adaptively resets or forgets the memory of the cell by scaling the internal state of the cell before adding it as input through the cell's self-recurring connection [23]. In this way, connections are formed by each cell adding the output and memory cell of the previous time step to the input of the next time step. Here, the output and memory cell from the previous time step are added to the input of the next time step.

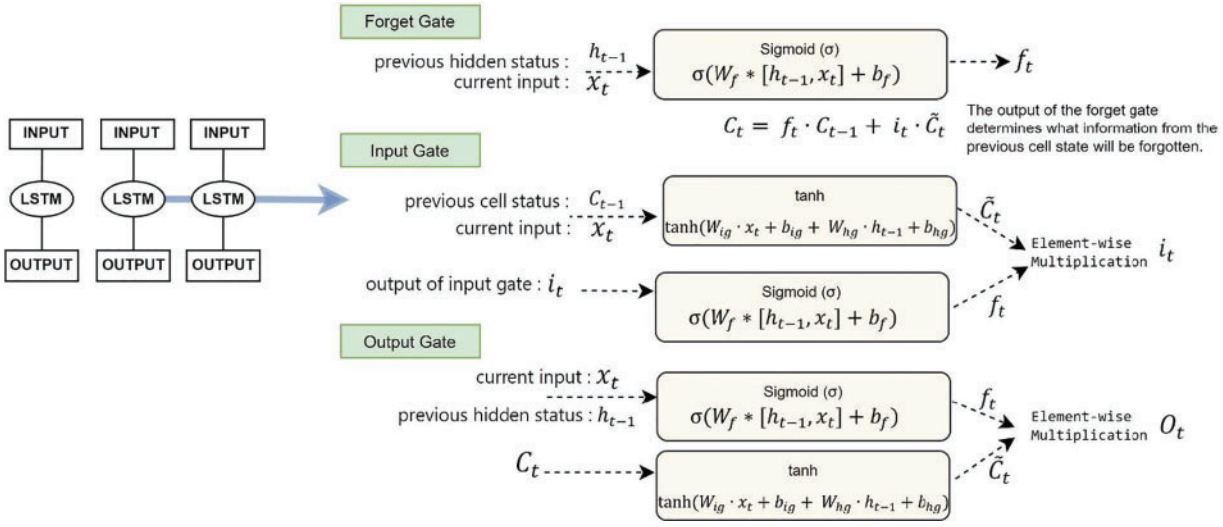

Figure 3: LSTM architecture

This allows the LSTM to have the capability to understand more context in the current time step by utilizing information from the previous time step. An LSTM network mathematical model is provided in Eqs. (1) to (6) [24].

$$i_t = \sigma\left(W^{(i)} \times (h_{t-1} \oplus x_t) + b^{(i)}\right) \tag{1}$$

In the Eq. (1), the output calculated by the input gate is represented. This output is used to determine the effect of input activations on the memory cell at the current time step of the network.

The term i_t represents the output of the input gate, σ is the sigmoid activation function, $\oplus$ denotes the vector concatenation, $W^{(i)}$ input weight matrix, x_t input vector, $b^{(i)}$ bias term for the input gate, h_{t-1} hidden state from the previous time step. Additionally, the sigmoid function in this process ensures that the output lies in the [0, 1] range, controlling the contribution of input activations to the memory cell.

$$f_t = \sigma\left(W^{(f)} \times (h_{t-1} \oplus x_t) + b^{(f)}\right) \tag{2}$$

In the Eq. (2), the output of the forget gate is calculated. So, it is determined which part of the memory cell in the current time step will be forgotten.

In this formula:

f_t: Forget gate output,

$W^{(f)}$: Hidden state weight matrix,

h_{t-1}: Hidden state in previous time step,

$b^{(f)}$: Bias term for forget gate

Following that, the portion of the current time step's hidden state to be conveyed to the output, and the part that will remain hidden, is determined using the formula described in Eq. (3).

$$o_t = \sigma\left(W^{(o)} \times (h_{t-1} \oplus x_t) + b^{(o)}\right) \tag{3}$$

o_t: Output gate output,

$W^{(o)}$: Input weight matrix for output gate,

$b^{(o)}$: Bias term for output gate

$$g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right) \tag{4}$$

Eq. (4) refers to a process in which the input vector (x_t), the hidden state in the previous time step (h_{t-1}) and the relevant weights are used to calculate the output of the memory gate.

The tanh function limits the output to the range [−1, 1], which determines which part of the memory gate is added to the memory cell.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{5}$$

In addition, the cell state is updated and the memory flow is controlled by first forgetting the previous cell state with the output of the forgetting gate ($f_t \odot c_{t-1}$), then adding the output of the input gate and the output of the memory gate ($i_t \odot g_t$), as shown Eq. (5).

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

Additionally, to update the hidden state, the hyperbolic tangent activation function in Eq. (6) is employed. The updated output of the hidden state, denoted as h_t, is represented by c_t for cell state output, and $\odot$ denotes element-wise (Hadamard) multiplication.

With the layer, a more comprehensive feature extraction is aimed by taking into account both past and future information at each time step.

4.4 Bidirectional Long Short-Term Memory

Bi-LSTM is a method developed based on the bidirectional operation of the LSTM architecture. The Bi-LSTM neural network consists of LSTM units operating bidirectionally to combine both past and future contextual information. The advantage of Bi-LSTM lies in its ability to learn long-term dependencies without storing repeated contextual information. In contrast to the LSTM network the Bi-LSTM network has two parallel layers that propagate in both directions, capturing dependencies in both contexts [25]. One layer processes the text from beginning to end (forward), while the other layer processes it from end to beginning (backward). The formulations for forward direction illustrated in from Eqs. (7) to (10).

$$i_{f,t} = \sigma\left(W^f_{ii}x_t + b^f_{ii} + W^f_{hi}h^f_{t-1} + b^f_{hi}\right) \tag{7}$$

$$f_{f,t} = \sigma\left(W^f_{if}x_t + b^f_{if} + W^f_{hf}h^f_{t-1} + b^f_{hf}\right) \tag{8}$$

$$o_{f,t} = \sigma\left(W^f_{io}x_t + b^f_{io} + W^f_{ho}h^f_{t-1} + b^f_{ho}\right) \tag{9}$$

$$g_{f,t} = \tanh\left(W^f_{ig}x_t + b^f_{ig} + W^f_{hg}h^f_{t-1} + b^f_{hg}\right) \tag{10}$$

Likewise, the backwards' illustrated from Eqs. (11) to (14).

$$i_{b,t} = \sigma\left(W^b_{ii}x_t + b^b_{ii} + W^b_{hi}h^b_{t-1} + b^b_{hi}\right) \tag{11}$$

$$f_{b,t} = \sigma\left(W^b_{if}x_t + b^b_{if} + W^b_{hf}h^b_{t-1} + b^b_{hf}\right) \tag{12}$$

$$o_{b,t} = \sigma\left(W^b_{io}x_t + b^b_{io} + W^b_{ho}h^b_{t-1} + b^b_{ho}\right) \tag{13}$$

$$g_{b,t} = \tanh\left(W^b_{ig}x_t + b^b_{ig} + W^b_{hg}h^b_{t-1} + b^b_{hg}\right) \tag{14}$$

Also, mathematical expression for updating cell state and hidden state showed in Eqs. (15)–(18), respectively.

$$c_{f,t} = f_{f,t} \odot c^f_{t-1} + i_{f,t} \odot g_{f,t} \tag{15}$$

$$c_{b,t} = f_{b,t} \odot c^b_{t-1} + i_{b,t} \odot g_{b,t} \tag{16}$$

$$h_{f,t} = o_{f,t} \odot \tanh\left(c_{f,t}\right) \tag{17}$$

$$h_{b,t} = o_{b,t} \odot \tanh\left(c_{b,t}\right) \tag{18}$$

The information obtained from both directions is then merged. This way, the dependence of a word on the words before and after it is analysed, preserving the context of the text.

4.5 Attention Mechanism

RNN-based networks such as CNN, LSTM, GRU, among others, have shown successful results in tasks related to text classification. However, these models may not be intuitive enough, especially when dealing with unexplainable misclassifications in reading hidden information. At this point, attention-based methods have been successfully employed in text classification tasks [9]. This technique allows the model to pay more "attention" to specific input elements, aiding in extracting important information more effectively.

Particularly when working with long or complex data, it is crucial for the model to focus only on specific sections. The Attention Mechanism accomplishes this focus. By helping the model to learn more flexibly and effectively, this mechanism enhances the performance of DL models.

4.6 Bidirectional Gated Recurrent Unit

The Bi-GRU is a type of bidirectional RNN that has only input and forget gates. It allows the use of both forward and backward details to make predictions about the current state, enabling a broader context understanding [26]. The working principle of Bi-GRU is as follows: in the forward pass, input data is processed forward through time steps, where at each step, the current input and information from the previous step are analysed by GRU cells. In the backward pass, input data is processed in the opposite direction to the previous one. At each step, the current input and information from the next step are evaluated. Subsequently, in the concatenation step, information from both forward and backward directions is combined. The formulation for these steps showed in from Eqs. (19) to (26). This allows us to obtain a representation at each step that includes both past and future context. This bidirectional approach enables Bi-GRU to have a better understanding of temporal context and model more complex relationships. Such a network can be particularly effective when working with data that changes over time or involves contextually rich information such as textual expressions.

For forward direction:

$$z_{f,t} = \sigma\left(W_{zf}[h^{f}_{t-1}, x_{zf}]\right) \tag{19}$$

$$r_{f,t} = \sigma\left(W_{rf}[h^{f}_{t-1}, x_{zf}]\right) \tag{20}$$

$$\tilde{h}_{f,t} = \tanh\left(W_{hf}[r_{f,t}h^{f}_{t-1}, x_{zf}]\right) \tag{21}$$

For forward direction:

$$z_{b,t} = \sigma\left(W_{zb}[h^{b}_{t-1}, x_{zb}]\right) \tag{22}$$

$$r_{b,t} = \left(W_{rb}[h^{b}_{t-1}, x_{zb}]\right) \tag{23}$$

$$\tilde{h}_{b,t} = \tanh\left(W_{hb}[r_{b,t}h^{b}_{t-1}, x_{zb}]\right) \tag{24}$$

For cell state update:

$$h_{f,t} = \left(1 - z_{f,t}\right) \odot h^{f}_{t-1} + z_{f,t} \odot \tilde{h}_{f,t} \tag{25}$$

$$h_{b,t} = \left(1 - z_{b,t}\right) \odot h^{b}_{t-1} + z_{b,t} \odot \tilde{h}_{b,t} \tag{26}$$

After the Bi-GRU layer, in proposed model, global max pooling layer was applied to select important values from the features obtained in the Bi-GRU layer, create the feature vector, reduce dimensionality and increase information density. Then, the dense layer is used especially in the final layers. And the layers are parametric layers in which weights and bias are updated during the learning process of the network [27].

4.7 Output

In the output layer, we used the 'SoftMax' function, which is commonly used for multi-classification problems. In the SoftMax function, model outputs are transformed into a probability distribution. The probabilities of the available outputs belonging to each class are calculated, as shown Eq. (27) [28]. The sum of probabilities for all classes should be equal to 1. The formula used for the SoftMax function in our study is as follows:

$$\text{SoftMax}\ (z)_i = \frac{e^{z_i}}{\sum_{j=1}^{3} e^{z_i}} \tag{27}$$

z_i: The input to the i-th neuron in the output layer

SoftMax $(z)_i$: The probability of the i-th class

5 Experiments

In this section, the data set used in the study and the performance evaluation metrics used to evaluate the model performance are mentioned.

5.1 Dataset

Within the scope of the study, we created a data set consisting of radiology article abstracts to identify texts written by AI and humans. The dataset contains approximately 54,000 samples. We obtained article abstracts using the following steps:

- First, we used the abstracts of articles written in the field of radiology and available on PubMed. PubMed is a free search engine that accesses the MEDLINE database, which consists primarily of references and abstracts on life sciences and biomedical topics. The United States National Library of Medicine (NLM) at the National Institutes of Health maintains the database as part of the Entrez retrieval system. By integrating an API provided by PubMed into the code we wrote in Python, we obtained approximately 18,000 radiology article abstracts. With these abstracts, we created the 'Human' class, which is the first class of our 3-class classification model.
- In the second stage, we identified 10 frequently used keywords in these abstracts with a code we wrote in Python, using the article abstracts we received from PubMed. We aim to enable GPT to produce new abstracts using the keywords we specify. For this purpose, we created a prompt that allows abstracts of at least 150 tokens to be created within the framework of keywords. In the next step, we wrote a Python code to ensure that this request was processed by the GPT-3.5-Turbo model, using an API provided by OpenAI. In this way, we produced 18,000 abstracts for the 'GPT' class, which is the third class of the classification.
- We can perform AI-generated and human-written text detection with these two classes. However, we created the 'Paraphrased' class to make the model more robust and to see whether it makes this distinction clearly. For this purpose, we manually entered the original article abstracts we received from PubMed into ChatGPT and had these examples rewritten by ChatGPT with a prompt we created, that is, human-written abstracts were paraphrased. Thus, we created 18,000 abstracts belonging to the 'Paraphrased' class, which is another class of our classification model.

The word/token distribution of the classes is shown in Fig. 4.

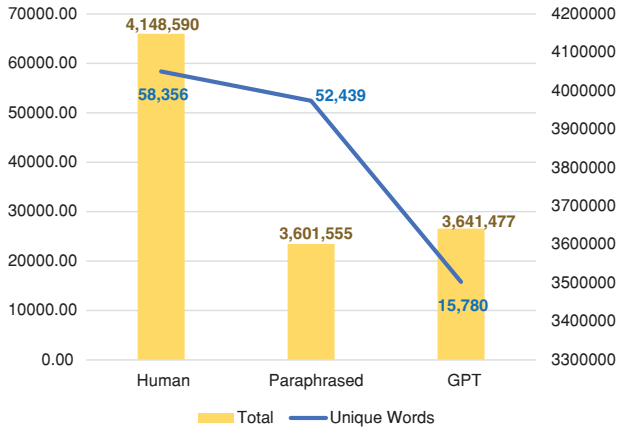

Figure 4: Distribution word counts of the classes

Consequently, we created a balanced dataset included the 'Human', 'Paraphrased' and 'GPT' classes and manually labelled the dataset. Statistical analysis and samples of labelled data of the dataset are listed in Tables 3 and 4, respectively.

Table 3: Statistical analysis of the dataset

	Entries	Human (%)	Paraphrased (%)	GPT (%)
Train set	42759	33.56	33.71	33.56
Validation set	5355	33.92	34.13	33.92
Test set	5345	34.44	32.95	34.44

Table 4: Classified samples of the dataset

Abstract	Class
Title: Does seed migration increase the risk of second malignancies in prostate cancer patients treated with Iodine-125 loose seeds brachytherapy? Purpose: Evaluate the risk of second malignancies after seed migration (MS) in prostate cancer patients treated with (125) I loose seeds brachytherapy. Methods and Materials: Data from 2802 prostate cancer patients treated with (125) I loose seeds brachytherapy in 3 Canadian centers. Postimplant pelvic radiography and CT scan performed for postimplant dosimetry to assess seed migration. Incidence of second malignancies determined through chart review. 7- and 10-year cumulative incidences of second malignancies calculated. Fine and Gray competing risk regression analysis used to assess factors associated with second malignancies. Results: Mean age: 63.5 years, $P = 0.510$). Advanced age was the only significant factor associated with second malignancies. Conclusions: Results suggest no increased risk of second malignancies associated with seed migration after (125) I loose seeds brachytherapy for prostate cancer. Longer follow-up and more events needed for a better correlation between seed migration and second malignancies.	Human
Title: Primary pulmonary diffuse large B-cell lymphoma associated with feline leukaemia virus infection in a young cat. Case Summary: A 4-year-old castrated male domestic shorthair cat with a continuous cough was examined. Thoracic radiography revealed a severe radiopaque region in the caudal lobe of the right lung. Subsequent CT at 108 days showed a 27 mm $\times$ 23 mm $\times$ 18 mm mass in the caudal lobe. No abnormalities were detected in other organs. Lobectomy was performed and the mass was diagnosed as diffuse large B-cell lymphoma associated with feline leukaemia virus (FeLV) infection. Immunohistochemistry showed positive staining for CD20 FeLV p27 and FeLV glycoprotein 70. Despite no signs of gastrointestinal or respiratory injury thickening in the jejunum wall was discovered 196 days post-lobectomy confirming high-grade lymphoma on fine-needle aspiration examination. Relevance and Novel Information: This case represents the first report of primary pulmonary diffuse large B-cell lymphoma associated with FeLV infection in a young cat. The findings contribute to the understanding of feline lymphoma and emphasize the potential involvement of FeLV in respiratory lymphoid neoplasia.	Paraphrased

(Continued)

Table 4 (continued)

Abstract	Class
Title: Radiological imaging method a comprehensive overview purpose. This paper provides an overview of the different forms of radiological imaging and the potential diagnosis capabilities they offer as well as recent advances in the field. Materials and Methods: This paper provides an overview of conventional radiography digital radiography panoramic radiography computed tomography and cone-beam computed tomography. Additionally recent advances in radiological imaging are discussed such as imaging diagnosis and modern computer-aided diagnosis systems. Results: This paper details the differences between the imaging techniques the benefits of each and the current advances in the field to aid in the diagnosis of medical conditions. Conclusion: Radiological imaging is an extremely important tool in modern medicine to assist in medical diagnosis. This work provides an overview of the types of imaging techniques used the recent advances made and their potential applications.	GPT

5.2 Experimental Settings

In this study, we developed a model to detect text generated by AI from texts written by humans. For this purpose, we address 3-classes classification problem using the dataset we created with radiology article abstracts. Our goal is to detect human-generated abstracts from GPT-generated, and also to detect human-generated abstracts that we paraphrase into the AI system. For this purpose, we built a hybrid model called 'HybridGAD' by combining deep neural network models (LSTM, Bi-LSTM, Bi-GRU) due to their success in NLP tasks. We also included the attention mechanism in our study in order to focus more on specific features representing classes.

Input\|Embedding: First of all, the input layer is used to create representations of textual expressions. For this aim we used Bio_ClinicalBERT tokenizer. The word representation obtained from Bio_ClinicalBERT were gave into embedding matrix for input. The embedding matrix used in the embedding layer was obtained with a domain-specific tokenizer (Bio_ClinicalBERT). In this way, word representations better represent context features, thus ensuring the robustness of the model.

Gaussian Noise: Afterwards embedding, a Gaussian Noise layer was applied to increase the resistance of the model. Gaussian Noise helps generalize the model. It is also aimed to prevent overfitting.

LSTM: LSTM layer has been added, which analyses the relationship between words based on context and obtains their high-quality features.

SpatialDropout1D: The features obtained from the LSTM layer are given as input to the SpatialDropout1D layer. The SpatialDropout1D layer helps reduce overfitting when working with sequential data (time series or text data).

Bi-LSTM: A more comprehensive feature extraction is aimed by taking into account both past and future information at each time step.

AttentionWithContext: With this layer, the model gives more weight to certain features. This helps the model be more effective in the learning process and better focus on specific contextual features.

Dropout: To prevent overfitting, a certain percentage of the neurons in the network are randomly turned off.

Bi-GRU: The layer added because it used to better understand contextual meaning.

Dropout: Dropout was added again to ensure generalization of the model and to prevent overfitting. GlobalMaxPooling1D: Pooling was performed to highlight important features on the data and help the model focus on these features.

Dense: A dense layer, which is a layer that creates outputs using weighted inputs, has been added.

Output: Finally, the output layer has been added, which converts the input data into class probabilities and predicts the class with the highest probability.

On the other hand, experiments were conducted with Word2Vec [29], commonly used in the field of NLP, to examine the differences between Bio_ClinicalBERT. The Word2Vec model, based on artificial neural networks and containing input, output, and hidden layers, is one of the widely used word embedding models. This model aims to transform words into continuous vector representations by utilizing the continuous bag-of-words (CBOW) [30] and skip-gram (SG) [31] algorithms to capture multiple similarity relationships between words [32].

Convolutional Neural Networks (CNNs) have increasingly gained prominence in various subfields of AI, thanks to their high performance in image processing. One such domain is NLP, where CNN models have demonstrated success in extracting features from words or textual expressions [23].

A fundamental CNN architecture, as showed in Fig. 5., typically takes as input a sequence of word vectors representing text documents. The subsequent layer, known as the embedding layer, is utilized to learn and represent these numerical vectors. If the input data is generated using a word embedding model, the embedding layer usually incorporates the weights of this pretrained model.

In cases where the input data is represented not by-word vectors but by-word indices, the embedding layer takes these indices and extracts the corresponding embedding vectors for each word. Next, convolutional operations are applied in the convolutional layer to extract features from the learned representations. This process involves convolution with a filter, sliding over the data. Different sets of filters continue the convolution process to search for higher-level combinations of defined features until features defining higher-level data are created in deeper layers. The subsequent step, the pooling layer, selects significant features from the output of the convolutional layer. The chosen features are combined in the fully connected layer and utilized for the final classification. In the output layer, the classification process of the model takes place [23].

We conducted the experiments using the hybrid models which include an embedding matrix as an input created using Word2Vec and hybrid models which consist of CNN, LSTM, Bi-LSTM, and Bi-GRU models. The models are CNN_AWC_BiLSTM, LSTM, CNN_BiLSTM_AWC, BiGRU_CNN, CNN_BiLSTM_AWC_BiGRU.

CNN_AWC_BiLSTM: In this hybrid model, CNN, attention mechanism and Bi-LSTM models are applied, respectively. First, spatial dropout was applied to the embedding matrix created with Word2Vec and then it was given as input to the convolution layer. Then, the pooling layer was added, and dimensional reduction was achieved. In the next step, the attention mechanism was applied and its outputs were given as input to the Bi-LSTM layer [27].

LSTM: In this model, the model was developed only with the LSTM architecture, and Word2Vec was used in the embedding layer. A 3-layer LSTM architecture was designed and a dropout was added between each layer.

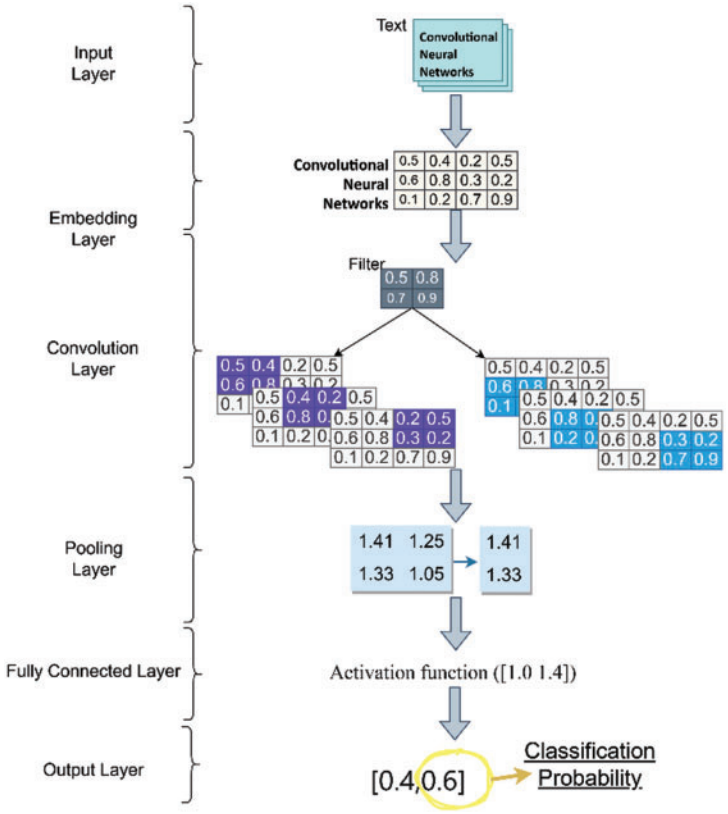

Figure 5: Example of the CNN architecture for text classification

CNN_BiLSTM_AWC: In this model, which is another hybrid model created similarly, the embedding matrix created with Word2Vec is first given as input to the CNN layer. The output obtained from the convolutional layer was given as input to Bi-LSTM in the next stage. A dropout was then implemented. The dropout process was followed by the attention mechanism. The outputs of the attention mechanism are transmitted to the output layer [27].

BiGRU_CNN: In the hybrid model created with Bi-GRU and CNN, after creating the embedding matrix with Word2Vec, GaussianNoise and SpatialDropout were applied, respectively. The resulting output was given as input to the Bi-GRU layer. Then, after the applied dropout, the convolutional layer was added.

CNN_BiLSTM_AWC_BiGRU: In this architecture created using four different methods, the convolutional layer was first added, and then the outputs of the convolutional layer were transmitted to the Bi-LSTM layer. Then, the features obtained with the applied attention mechanism were transmitted to the Bi-GRU layer in the last layer [27].

All the experiments carried out in the study were developed on Google Colab. We divided the dataset into 80% training and 20% test samples, and used half of the test data for validation. In the proposed model, we stated that the data size that will feed the model in the input layer can be variable instead of a fixed value. We set the Gaussian Noise to a value of 0.2 to avoid overfitting. It is the embedding layer that converts tokens into digital vectors. 100 represents the embedding size. We initiated our model with these values. The values of other layers are shown in Fig. 6. In addition to Fig. 6, the count of parameters of the model is 215,043 and the size of the model (in bytes) is

860,172. Additionally, we set the batch size and epoch values to 32 and 8, respectively. Besides, we set the maximum word length as 150 in the tokenizer process and the learning rate as '1e−3' during the model construction. Additionally, we calculated the following metrics to evaluate the computational burden of HybridGAD model. The prediction time and model memory usage are measured as 0.70 s and 48 bytes, respectively. These metrics provide valuable information about our model's processing time and memory usage. The fact that the prediction time is 0.70 s shows that the model works fast and can be used effectively in real-time applications. Moreover, considering that the model uses only 48 bytes of memory, it seems that it can work effectively even on devices with low system resources. These metrics highlight the usability of the HybridGAD model in practical applications.

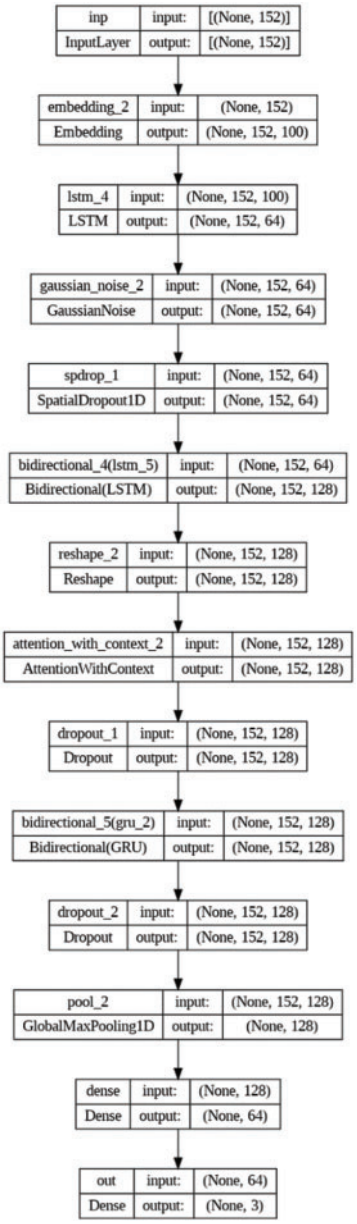

Figure 6: Summary of HybridGAD model

In addition to addressing a deficiency in existing studies, our study also touches upon a point that will detect texts written by humans and rewritten (paraphrased) by generative AI. In other words, it not only performs human-written and AI-generated detection but also detects paraphrased texts. On the other hand, online detectors have been used within the scope of similar studies to distinguish AI-generated and human-written texts.

Although these detectors generally performed well, they did not show high performance in classifying the academic articles in our data set. We conducted a study to detect who wrote the 50 examples in our data set with 55 different online detectors [33–37], and we shared the accuracy values of the results we obtained in Table 5.

Table 5: Accuracy values of online AI-generated text detector

AI content detector	Acc. (%)		
	Human	Paraphrased	GPT
Duplichecker [33]	1	0.04	0.02
Sapling [34]	1	0.88	0.16
ZeroGPT [35]	0.44	0.42	0.42
Quillbot [36]	1	0.62	0.12
GPTZero [37]	1	0.04	0.32

According to the result in Table 5, although online detectors generally performed well, they performed poorly when we tested with our radiology abstract dataset. The accuracy values in Table 5, were obtained by dividing the sum of correctly classified samples by the total number of samples. Experiments were conducted for some of the samples in the dataset. The percentage values given by online detectors (rate of produced by AI) are considered correct if they are between 0%–50% for the 'Human' class. For the 'Paraphrased', it is considered correct if it is between 50%–80%. Because online detector could not label it as either 'Human' or 'GPT'. Finally, for the 'GPT' class, a score between 75%–100% was considered a correct prediction. The ratio of correct predictions to all predictions is defined as the accuracy value.

5.3 Evaluation Metrics

In the current study, we used several evaluation metrics frequently used in NLP to evaluate model performance. These are accuracy, F1-score, Precision, MCC, Kappa, ROC-AUC and Perplexity metrics.

Accuracy: It can be defined as the ratio of correct predictions to total predictions, simply. That is, it is calculated as the ratio of the number of examples correctly classified by the model to the number of all predictions.

Precision: The metric measures the rate of examples of the positive class of the model actually belonging to the positive class.

Recall: The metric evaluates how successfully the model detects examples that correctly belong to the positive samples.

F1-score: The metric used to evaluate performance, especially in studies with unbalanced data sets. Because it takes the harmonic average of precision and recall metrics in such data sets, it prevents these metrics from overlapping each other.

MCC (Matthews Correlation Coefficient): It provides a measure of performance by taking into account correct and incorrect predictions in the classification results.

Kappa: The Cohen's Kappa metric is a statistical measure that measures how well the model's predictions agree with the actual values, taking into account the imbalance between classes in classification problems.

ROC-AUC: ROC curve is a graph that shows the relationship between sensitivity and specificity of a model. ROC-AUC refers to the area under the ROC curve.

Perplexity: Perplexity measures how accurate and consistent a language model's predictions are over a text. A lower perplexity indicates that the model performs better on text, meaning it makes less inconsistent predictions.

If these all metrics are close to 1, it means that the model performance is high.

5.4 Result and Discussion

The obtained results illustrate the successful ability of the proposed model to classify human generated, paraphrased texts, and GPT generated texts. The high accuracy of the model indicates its effective capacity to recognize fundamental differences between texts generated by AI and human writing. This suggests that the model exhibits a robust ability to understand and analyses various features of language. The prediction performance of the proposed HybridGAD model was evaluated using most common eight performance evaluation metrics such as accuracy, F1-score, Precision, Recall, MCC, Kappa, ROC-AUC, and Perplexity. For a dataset containing approximately 54,000 samples (80% training, 20% testing samples), the HybridGAD model achieved 98% accuracy, 97% F1-score, 97% precision, 97% recall, 96% MCC, 96% Kappa, 98% ROC-AUC, and a Perplexity value of 1.0455.

As a second phase, five different hybrid deep neural network models explained in Section 4.2. were examined. Word2Vec was used as the tokenizer in all models. The performance results of the models are shown in Table 6. Upon examining the table, we observed that the CNN_AWC_BiLSTM hybrid architecture model, created by combining CNN, attention mechanism, and Bi-LSTM models, is the model closest to our HybridGAD model in terms of performance. The CNN_AWC_BiLSTM model achieved similar results to our model in terms of accuracy, F1-score, and Precision metrics, but it performed below our model in the other five metrics. In the other hybrid models, the second-highest accuracy value, 95%, was observed in the BiGRU_CNN hybrid model consisting of Bi-GRU and CNN layers.

Table 6: Performance result of the models using Word2Vec

Model	Embedding	Acc.	F1	Precision	Recall	MCC	Kappa	ROC	Perplexity
CNN_AWC_BiLSTM	Word2Vec	0.98	0.97	0.97	0.96	0.95	0.95	0.97	1,0728
BiGRU_CNN	Word2Vec	0.95	0.92	0.93	0.92	0.89	0.89	0.94	1,0336
CNN	Word2Vec	0.93	0.89	0.93	0.85	0.84	0.84	0.91	1,2742
BiLSTM_CNN	Word2Vec	0.93	0.89	0.90	0.89	0.84	0.84	0.92	1,2196

(Continued)

Table 6 (continued)

Model	Embedding	Acc.	F1	Precision	Recall	MCC	Kappa	ROC	Perplexity
CNN_BiLSTM_AWC_BiGRU	Word2Vec	0.92	0.87	0.90	0.87	0.82	0.81	0.91	1,1895
CNN_BiLSTM_AWC	Word2Vec	0.91	0.85	0.86	0.86	0.79	0.78	0.89	1,5336
LSTM	Word2Vec	0.90	0.84	0.85	0.84	0.77	0.76	0.88	1,4605
HybridGAD (Our proposed model)	**Bio_Clinical BERT**	0.90	0.84	0.85	0.84	0.77	0.76	0.88	**1,0455**

In conclusion, when compared with the proposed HybridGAD model, it is evident that the HybridGAD model exhibits the highest performance, surpassing all five models.

As a result, the developed classification model stands out with its effective ability to understand different features and structures of language. This study sheds light on future research by providing a new approach to understand the complexity of language and accurately classify texts produced by AI. Additionally, when we conducted a class-wise evaluation for the proposed model, examining metric values for each class, we observed remarkably successful results. Particularly, in identifying abstracts belonging to the 'Human' class, the model achieved significantly high performance. This indicates that the model is successful in detecting texts written by humans, showcasing its proficiency in this aspect. The results are shown in Fig. 7.

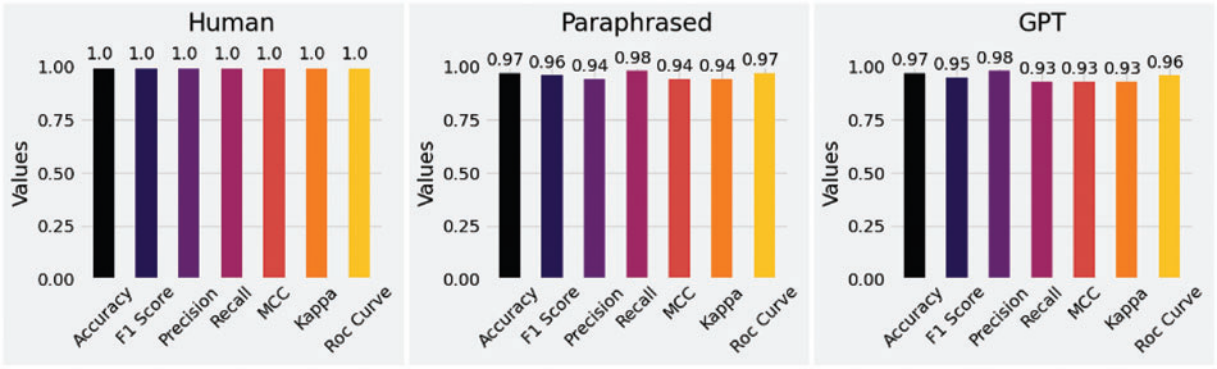

Figure 7: Class-wise performance results of HybridGAD model

6 Ablation Study

We now also considered an ablation study to investigate how classification performance would be affected when using glove as a word representation. Contrary to Word2Vec, which depends solely on nearby information within a local context window, the GloVE algorithm incorporates word co-occurrence data and global statistics. In GloVE, global matrix factorization is employed, representing the presence or absence of words in a document. GloVE operates as a log-bilinear model, commonly known as a count-based model. We also tried a matrix created with glove (300 dim) on the 5 models we created with the embedding matrix Word2Vec technique. We left all other parameters constant to measure only the impact of the GloVE. Experimental results for 5 models are shown in Table 7.

Table 7: Performance result of the models with GloVE

Model	Embedding	Acc.	F1	Precision	Recall	MCC	Kappa	ROC	Perplexity
CNN_BiLSTM_AWC	GloVE	0.98	0.97	0.97	0.97	0.95	0.95	0.98	1,4610
LSTM	GloVE	0.98	0.96	0.96	0.96	0.94	0.94	0.97	1,1285
CNN_AWC_BiLSTM	GloVE	0.96	0.94	0.94	0.94	0.91	0.91	0.96	1,5441
BiGRU_AWC_CNN	GloVE	0.95	0.93	0.94	0.93	0.90	0.90	0.95	1,1506
BiGRU_CNN	GloVE	0.94	0.90	0.91	0.90	0.86	0.85	0.93	1,2186
CNN_BiLSTM_AWC _BiGRU	GloVE	0.88	0.80	0.86	0.81	0.74	0.72	0.86	1,2454
CNN_GRU	GloVE	0.70	0.49	0.61	0.52	0.36	0.31	0.66	5,1567
HybridGAD (Our proposed model)	**Bio_Clinical BERT**	**0.98**	**0.97**	**0.97**	**0.97**	**0.96**	**0.96**	**0.98**	**1,0455**

Firstly, when comparing the experimental results with Word2Vec, it can be stated that the CNN_BiLSTM_AWC model, i.e., the hybrid model constructed with CNN first, followed by Bi-LSTM, and then the attention mechanism, has achieved results closest to our proposed model. However, especially the perplexity value is significantly lower than our model. The model constructed with LSTM has the second-highest accuracy value, but when looking at other metrics, it has not reached the performance of our proposed model.

When comparing the results of GloVE and Word2Vec, it is clearly seen that there are differences between the results. Although the experiments with GloVE have low performance from our model, they have shown better classification performance compared to the experiments with Word2Vec. All experiments conducted in the study, and their results are presented in Figs. 8–10.

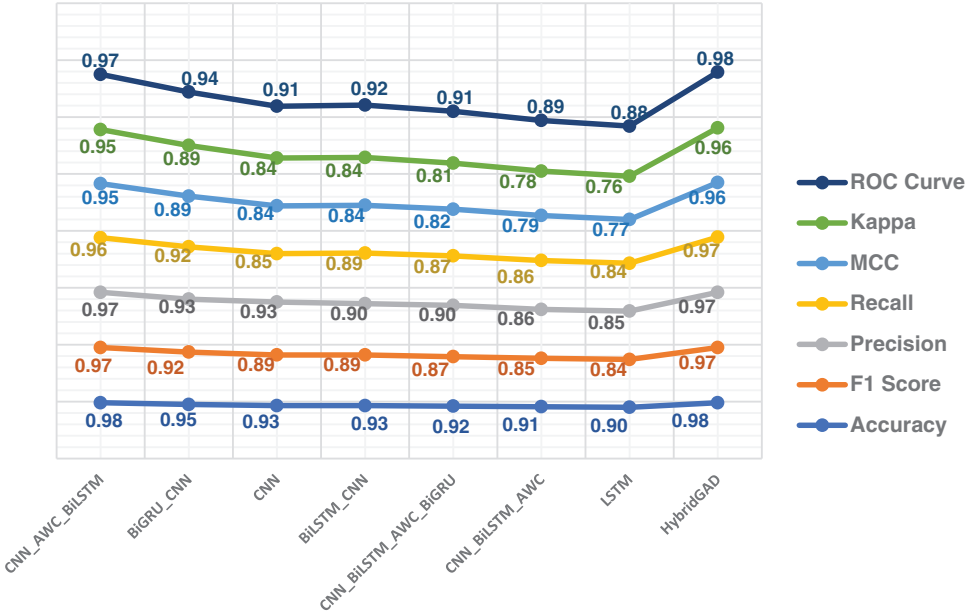

Figure 8: Performance results of all the models using Word2Vec

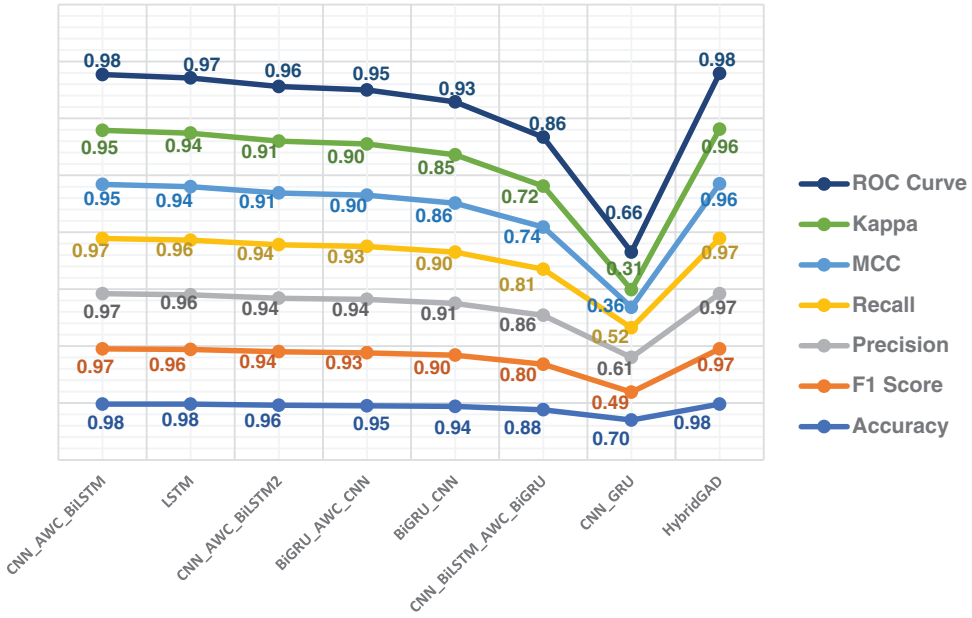

Figure 9: Performance results of all the models using GloVE

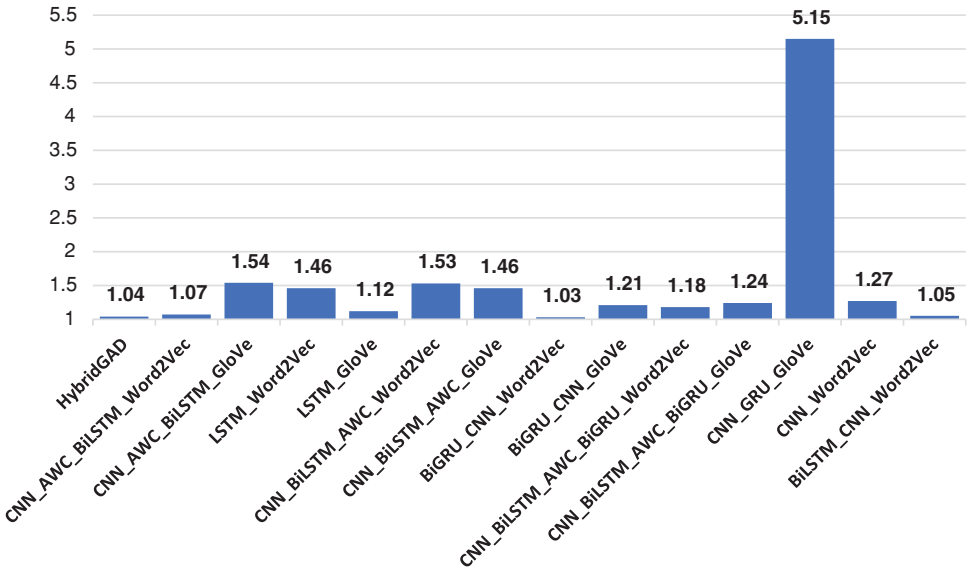

Figure 10: Perplexity values of all the models

Also, we conducted another experiment to prove the performance of our proposed model, HybridGAN. In this experiment, we discussed the components (LSTM, Bi-LSTM, Bi-GRU and attention mechanism) that make up our hybrid model one by one. First, we trained our dataset with LSTM and measured the prediction performance of the model. Then we continued the process with LSTM and Bi-LSTM. In both experiments, we reached similar results with an accuracy value of 66%. As the next step, we examined the prediction performance of the Bi-GRU model. The accuracy value we obtained only with Bi-GRU was measured as 67%. Finally, we evaluated the contribution of the

attention mechanism to the HybridGAN. We removed the attention mechanism from HybridGAN model and obtained the same accuracy value approximately 67%.

The experimental results we obtained showed that models are more performant when we design them in combination, rather than modelling them individually. This has proven that using the components in the HybridGAN model together is more convenient and robust.

7 Conclusion

Detection of texts generated by AI is a subject of increasing importance today. With developing technology, the text generation capabilities of AI-based systems have increased significantly. However, these developments have also brought with them serious problems such as fake news, manipulative content and information pollution. The effects of such texts on societies can be profound and far-reaching, making it difficult to access reliable information and negatively affecting public debate. Therefore, correctly identifying the source of texts is a fundamental step to increasing reliability in accessing information and combating information pollution. This study aimed to develop a model that classifies texts generated by humans and AI models, focusing especially on the detection of texts generated by AI. Moreover, the inclusion of the class containing paraphrased texts increased the complexity of the model and provided a more sensitive approach to distinguishing AI-generated texts from human-generated. Thus, by increasing the reliability of the information resources provided to society, a healthy environment can be created for accessing information. The resulting three-class model is capable of successfully distinguishing human, paraphrased, and GPT-generated texts. This achievement highlights its ability to recognize key differences between AI-generated texts and human writing. Moreover, the results show that such models make significant progress in the ability to understand the subtleties and structures of language. In this context, in the future, it is important to explore new techniques to focus on more complex language structures and increase the sensitivity of the model to accelerate developments in this field. Also, with a prediction time of only 0.70 s and a relatively low memory usage of 48 bytes, the model provides efficiency in computational resources, making it suitable for real-time applications. If the model proves effective across different datasets and scenarios, it increases its generalizability potential and increases its usability in a variety of contexts. Conversely, the inclusion of paraphrased text and the need to distinguish between various text sources can increase the complexity of the model, making it more difficult to interpret and maintain.

It is thought that this model can make a significant contribution to real-world applications in detecting fake news or manipulative content. In this way, it can be used as an effective tool to ensure information security and increase social welfare. In addition, we aim to build a more comprehensive and more generalizable model with different data types by developing a graph-based model.

Acknowledgement: Not applicable.

Funding Statement: The authors confirm that this study did not receive any specific funding.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Aytuğ Onan; data collection: Tuğba Çelikten; analysis and interpretation of results: Aytuğ Onan; draft manuscript preparation: Tuğba Çelikten. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

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

References

[1] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2098–2117, May 2022. doi: 10.1016/j.jksuci.2022.02.025.

[2] Q. Li *et al.*, "A survey on text classification from traditional to deep learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–41, Mar. 2022. doi: 10.1145/3495162.

[3] Y. Lu, X. Zhao, and J. Wang, "Medical knowledge-enhanced prompt learning for diagnosis classification from clinical text," in *Proc. 5th Clin. Nat. Lang. Process. Workshop*, Toronto, Canada, Jul. 2023, pp. 278–288.

[4] A. Akram, "An empirical study of AI generated text detection tools," arXiv preprint arXiv:2310.01423, Oct. 2023.

[5] E. Alsentzer *et al.*, "Publicly available clinical BERT embeddings," arXiv preprint arXiv:1904.03323, Apr. 2019.

[6] V. Verma *et al.*, "Ghostbuster: Detecting text ghostwritten by large language models," arXiv preprint arXiv:2305.15047, May 2023.

[7] E. Mitchell *et al.*, "DetectGPT: Zero-shot machine-generated text detection using probability curvature," arXiv preprint arXiv:2301.11305, Jan. 2023.

[8] I. Cingillioglu, "Detecting AI-generated essays: The ChatGPT challenge," *Int. J. Inf. Learn. Technol.*, vol. 40, no. 3, pp. 259–268, Mar. 2023. doi: 10.1108/IJILT-03-2023-0043.

[9] K. Ibrahim, "Using AI-based detectors to control AI-assisted plagiarism in ESL writing: 'The terminator versus the machines'," *Lang. Test. Asia.*, vol. 13, no. 1, p. 46, Jan. 2023. doi: 10.1186/s40468-023-00260-2.

[10] S. C. Lee *et al.*, "Feature analysis for detecting mobile application review generated by AI-based language model," *J. Inf. Process. Syst.*, vol. 18, no. 5, pp. 1–14, May 2022.

[11] S. Sadiq, T. Aljrees, and S. Ullah, "Deepfake detection on social media: Leveraging deep learning and FastText embeddings for identifying machine-generated tweets," *IEEE Access*, vol. 11, pp. 95008–95021, 2023.

[12] A. F. Oketunji, "Evaluating the efficacy of hybrid deep learning models in distinguishing AI-generated text," arXiv preprint arXiv:2311.15565, 2023.

[13] R. Gaggar, A. Bhagchandani, and H. Oza, "Machine-generated text detection using deep learning," arXiv preprint arXiv:2311.15425, 2023.

[14] A. Gambetti and Q. Han, "AiGen-FoodReview: A multimodal dataset of machine-generated restaurant reviews and images on social media," arXiv preprint arXiv:2401.08825, 2024.

[15] F. Harrag, M. Debbah, K. Darwish, and A. Abdelali, "BERT transformer model for detecting Arabic GPT2 auto-generated tweets," arXiv preprint arXiv:2101.09345, 2021.

[16] I. Katib *et al.*, "Differentiating chat generative pretrained transformer from humans: Detecting ChatGPT-generated text and human text using machine learning," *Mathematics*, vol. 11, no. 15, pp. 3400, 2023. doi: 10.3390/math11153400.

[17] M. Gambini, M. Aveenuti, F. Falchi, M. Tesconi, and T. Fagni, "Detecting generated text and attributing language model source with fine-tuned models and semantic understandingunderstanding" in *IberLEF 2023*, Jaén, Spain, 2023.

[18] A. M. Hopkins, J. M. Logan, G. Kichenadasse, and M. J. Sorich, "Artificial intelligence chatbots will revolutionize how cancer patients access information: ChatGPT represents a paradigm-shift," *JNCI Cancer Spectrum*, vol. 7, no. 2, pp. pkad010, 2023. doi: 10.1093/jncics/pkad010.

[19] L. Hickman *et al.*, "Text preprocessing for text mining in organizational research: Review and recommendations," *Organ. Res. Methods.*, vol. 25, no. 1, pp. 114–146, 2022. doi: 10.1177/1094428120971683.

[20] Y. Gu *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Trans. Comput. Healthc.*, vol. 3, no. 1, pp. 1–23, 2020.
[21] B. Clavié and K. Gal, "EduBERT: Pretrained deep language models for learning analytics," arXiv preprint arXiv:1912.00690, 2019.
[22] G. K. Shrivastava, R. K. Pateriya, and P. Kaushik, "An efficient focused crawler using LSTM-CNN based deep learning," *Int. J. Syst. Assurance Eng. Manag.*, vol. 14, no. 1, pp. 391–407, 2023. doi: 10.1007/s13198-022-01808-w.
[23] B. Jang *et al.*, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, pp. 5841, 2020. doi: 10.3390/app10175841.
[24] F. Li, M. Zhang, G. Fu, T. Qian, and D. Ji, "A Bi-LSTM-RNN model for relation classification using low-cost sequence features," arXiv preprint arXiv:1608.07720, 2016.
[25] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, "Sentiment analysis using gated recurrent neural networks," *SN Comput. Sci.*, vol. 1, no. 2, pp. 1–13, 2020. doi: 10.1007/s42979-020-0076-y.
[26] Z. Li, S. H. Wang, R. R. Fan, G. Cao, Y. D. Zhang and T. Guo, "Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling," *Int. J. Imaging Syst. Technol.*, vol. 29, no. 4, pp. 577–583, 2019. doi: 10.1002/ima.22337.
[27] M. Kamyab, G. Liu, A. Rasool, and M. Adjeisah, "ACR-SA: Attention-based deep model through two-channel CNN and Bi-RNN for sentiment analysis," *PeerJ Comput. Sci.*, vol. 8, no. 4, p. e877, 2022. doi: 10.7717/peerj-cs.877.
[28] P. Song, C. Geng, and Z. Li, "Research on text classification based on convolutional neural network," in *2019 Int. Conf. Comput. Netw., Electron. Autom. (ICCNEA)*, Xi'an, China, 2019, pp. 229–232.
[29] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Computing numeric representations of words in a high-dimensional space," *U.S. Patent No. 9*, vol. 38, p. 464, 2015.
[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Adv. Neur. Inf. Process. Syst.*, vol. 26, pp. 3111–3119, 2013.
[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR Workshop*, 2013.
[32] A. Onan, "Sentiment analysis in Turkish based on weighted word embeddings," in *2020 28th Signal Proc. Commun. App. Conf. (SIU)*, Gaziantep, Turkey, 2020, pp. 1–4.
[33] "Duplichecker," Accessed: Apr. 20, 2024. [Online]. Available: https://www.duplichecker.com/
[34] "Sapling," Accessed: Apr. 20, 2024. [Online]. Available: https://sapling.ai/ai-content-detector
[35] "ZeroGPT," Accessed: Apr. 20, 2024. [Online]. Available: https://www.zerogpt.com/
[36] "Quillbot," Accessed: Apr. 20, 2024. [Online]. Available: https://quillbot.com/ai-content-detector
[37] "GPTZero," Accessed: May 22, 2024. [Online]. Available: https://gptzero.me/