IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 March 2025, accepted 10 April 2025, date of publication 21 April 2025, date of current version 30 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3562750

RESEARCH ARTICLE

Detection of AI-Generated Texts: A Bi-LSTM and Attention-Based Approach

JOHN BLAKE[1], (Member, IEEE), ABU SALEH MUSA MIAH[1], (Member, IEEE), KRZYSZTOF KREDENS[2], AND JUNGPIL SHIN[1], (Senior Member, IEEE)

[1]School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[2]Aston Institute for Forensic Linguistics, Aston University, B4 7ET Birmingham, U.K.

Corresponding authors: John Blake (jblake@u-aizu.ac.jp), Abu Saleh Musa Miah (musa@u-aizu.ac.jp), and Jungpil Shin (jpshin@u-aizu.ac.jp)

This work was supported in part by Seed-Corn funding received from the Aston Institute for Forensic Linguistics.

ABSTRACT This paper presents a novel algorithm that leverages cutting-edge machine-learning techniques to accurately and efficiently detect AI-generated texts. Rapid advancements in natural language processing models have led to the generation of text closely resembling human language, making it increasingly difficult to differentiate between human and AI-generated content. However, misuse of such texts presents a serious and imminent threat to the quality of academic publishing. This underscores the urgent need for robust detection mechanisms to ensure information quality, maintain trust, and preserve the integrity of research publications. Our proposed model outperformed existing algorithms for accuracy with less computational complexity. The proposed model is a feature-based hybrid deep learning network that leverages part-of-speech tagging and integrates Bidirectional Long Short-Term Memory (Bi-LSTM) networks with Attention modules. The initial module extracts local contextual features using convolutional layers, followed by Bi-LSTM layers that capture long-term dependencies from past and future sequences. An attention mechanism highlights critical sequence components, enhancing the model's focus on relevant data. The outputs from the attention and initial modules are concatenated through a residual connection, ensuring comprehensive feature representation. This combination is then fed into dense layers for final classification, effectively balancing feature richness and computational efficiency. The proposed model was evaluated on two benchmark datasets, achieving 85.00% and 88.00% accuracy, respectively.

INDEX TERMS AI-generated text detection, authorship analysis, authorship verification, machine-generated text detection.

I. INTRODUCTION

Radical innovations, such as artificial intelligence, big data and robotics, may be classed as disruptive technologies [1], overturning established models, systems or practices. Large Language Models (LLMs) are the latest disruptive technology. LLMs can not only read, paraphrase, and simplify digital texts but also generate human-like texts. The transformer model [2] marked a major milestone in the domain of deep learning and Natural Language Processing (NLP). Since its creation, various forms of state-of-the-art (SOTA) transformer models, such as the Generative Pre-Training model (GPT) [3], Bidirectional Encoder Representations from Transformers (BERT) [4] and Transformer-XL [5], have been introduced and utilized for a wide range of NLP tasks. Transformer models have produced outstanding results in many domains and on many tasks, including *inter alia* Natural Language Generation (NLG) [6], text classification [7], [8], machine translation [9], and text summarization [10].

Powerful LLMs, such as ChatGPT [11], can solve complex problems, write essays and create short answers in response to assignment questions. Until now, short answers and essays have served as appropriate assessment instruments for school teachers and university faculty to evaluate students' ability to draft logical, evidence-based answers. Conventional

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

plagiarism in such assignments could be checked using similarity detection tools, such as iThenticate for research articles and Turnitin for academic essays [12], but these plagiarism detection systems were not designed to detect AI-generated content, and so currently perform poorly at detecting texts created by LLMs.

LLMs are now able to create highly technical and scientific research articles, which, to a lay audience, may be indistinguishable from those written by renowned domain experts. Success in academia is gained primarily through publishing in top-tier journals, and since most top-tier journals use English as the *lingua franca*, this means ''publish (in English) or perish'' [13]. However, the advent of LLMs means that non-specialists can now generate generically appropriate scientific articles [14], and so non-experts can simply input a few prompts to an LLM and create a research article. However, given that AI-generated articles may contain factual inaccuracies, such as hallucinations, and non-specialists lack the domain knowledge needed to verify the accuracy of the content of the generated texts, this may lead to a proliferation of error-ridden research studies, which, in turn, places a higher burden on the academic gatekeepers, i.e. editors and reviewers, of academic journals; and may result in a devaluing of the authority of experts. Until now, the authors shared the ethical responsibility jointly and severally for the veracity of the content of scientific papers. Axiomatically, LLMs are unable to share such responsibility, which explains why many publishers have announced policies banning the attribution of authorship to LLMs. For example, Springer Nature declared that ''attribution of authorship carries with it accountability for the work, which cannot be effectively applied to LLMs'' [14].

A. RESEARCH NICHE

Cabanac and Labbé [15] discovered numerous scientific articles generated using outdated textual generation models, such as SciGen1. These models harness Context-Free Grammar (CFG), which tends to produce nonsensical, incoherent paragraphs and phraseologies that substantially deviate from expected or unmarked usage [16], exhibiting low vocabulary richness and high degrees of markedness. The weirdly paraphrased versions of scientific terms created by such models have been termed ''tortured phrases'' [17]. As such, texts generated by CFGs are relatively easy to automatically detect through intertextual distance and automatic clustering [18].

In contrast, newer text generation models based on LLMs, such as the GPT series, have significantly improved the quality of AI-generated texts. For example, the GPT-2 model [6] can produce texts that closely resemble those written by humans [19] while the performance of GPT-4 released by OpenAI in 2023 surpasses previous releases and displays ''human-like performance'' [20]. When used appropriately and ethically, the automatic generation of texts may have a beneficial role in the article creation process, and streamline the writing-up process. There is, however, cause for concern regarding potential risks associated with the misuse of such models by less scrupulous users. For example, these models may be misused for malicious tasks, such as fake news generation [21], [22], fake review generation [23], and viral story generation [24], [25]. In academia, these models have been applied to automatically generate research articles [26], reviews [27], and theses [28]. AI-generated research articles pose a serious threat to the research community. Readers of research articles may doubt their authenticity; editors and reviewers do not just have to check the novelty, substance, rigour and significance of research articles but also need to ascertain whether articles were AI-generated, AI-enhanced, or human-created [29], [30].

Therefore, there is an urgent need for a tool that can detect AI-generated academic texts. Nguyen and Labbé demonstrated the importance of detecting and removing nonsensical AI-generated papers from the scientific domain [31]. They identified a huge number of nonsensical CFG-generated articles that were published in various reputed journals. They also explain that AI-generated content is problematic for the scientific community and adds bias to publication metrics of journals (e.g. impact factor) and researchers (e.g. h-index). A central problem is that these papers interweave factual and fake information, adopting coherent structures incorporating tables, figures, and formulas; thus giving the appearance of *bonafide* articles, which may mislead inexperienced researchers.

In addition, such biased publication metrics and publications of fake results have appeared in scientific journals and conference proceedings in well-known bibliographic indexes such as Web of Science, Scopus, and Google Scholar. Moreover, IEEE and Springer withdrew more than 120 papers because of the AI-generated content, figures and fake tables [18]. The dissemination of disinformation or misinformation harms both the scientific community and society in general, resulting in a loss of trust due to the lack of faith in the findings of research and increasing scepticism on scientific progress. Thus, AI-generated papers can potentially cause significant harm to the scientific community and society as a whole. To maintain high standards of integrity and transparency, it is essential to prevent the publication of such fraudulent research. An effective and efficient detection method for AI-generated texts is needed to protect both scientists and society.

B. RESEARCH CONTRIBUTION

Although advances in developing AI-generated text detection approaches have been made in recent years, current systems exhibit several drawbacks and limitations. These include poor performance accuracy and limited contextual understanding. While many text detection methods can identify individual words or characters, they may be unable to grasp the meaning in context. This can be a limitation in situations where context is important for accurate analysis. Another notable weakness is the necessity for high computational complexity, which must be addressed to improve the accuracy and usefulness of

a detection system. To address these challenges, we propose a novel part-of-speech (POS) tagging-based bidirectional Long Short-Term Memory (LSTM) and attention mechanism to detect AI-generated texts. This method can increase the contextual understanding ability of the model and increase the performance efficiency while maintaining accuracy.

Rule-based models were first applied to extract POS tagging in the study. Based on the POS tagging features, a deep learning model was built using the Keras library, tailored for sequence classification tasks. The model architecture includes various enhancements for optimized performance. Specifically, a feature-based hybrid deep learning network was integrated that combines Bi-LSTM networks with an attention mechanism.

The input to the model is a sequence of integers of length `maxlen` representing tokens in a text. These tokens are embedded into dense vectors via an embedding layer. The embedded sequence is processed through a bidirectional LSTM layer with 64 units, enabling the model to capture dependencies from both past and future contexts, which allows the model to learn from the sequence in both forward and backward directions. To refine the focus of the model on significant elements within the sequence, an attention mechanism was incorporated to highlight critical components. In addition, the initial module, which consists of convolutional layers, extracts local contextual features. The outputs from the attention mechanism and the initial module are concatenated using a residual connection, ensuring a comprehensive feature representation that enhances the model's learning capability. This combination is fed into the classification module, balancing computational efficiency and feature richness.

This architecture demonstrated strong performance when evaluated on benchmark datasets, achieving accuracy scores of 85.00 % and 88.00%, confirming its robustness and superiority over existing methods in terms of accuracy and computational cost.

The remainder of the paper is organized as follows. Section II provides the literature review of the related works in the field of both human and automated detection of AI-generated texts. Section III introduces the datasets sourced for the evaluation of our model. Section IV details the key aspects of the proposed model, while Section V presents the results of the evaluation. This is followed by a conclusion, which briefly summarizes the contributions of this research and identifies future work.

II. RELATED WORKS

AI-generated texts may be detected by humans or automated systems, both of which have benefits and drawbacks. The primary drawback of human detection is the lack of scalability while drawbacks in automated detection include accuracy, reliability and explainability. It should be noted that these three aspects also impinge on human detection to varying degrees.

A. HUMAN DETECTION OF AI-GENERATED TEXT

Many researchers have investigated human detection of AI-generated content. Liyanage et al. [32] visualized different techniques to detect malicious or fake scientific texts. This work used two artificially generated research articles: a partial text substitution and a completely synthetic text. Bakhtin et al. [33] proposed an auto-regressive model to distinguish human-created text from AI-generated text. They claimed their model achieved more sensitivity compared to the existing models. Ippolito et al. [34] demonstrated that based on semantic errors, human detectors could detect AI-generated text easily with high performance accuracy without the assistance of any tools. To investigate this, they considered different factors, such as sampling techniques for the length of the text excerpt. The main drawback of these human detection models is that they only use the length of the sentence and semantic error. Gehrmann et al. [30] proposed a Giant Language model Test Room (GLTR) tool for human detectors to detect the AI-generated text with statistical techniques. They claimed that human detectors could improve the detection of artificially generated content from 54% to 72%. Dugan et al. [35] demonstrated that AI-generated fake texts could deceive the human detector for two or more sentences with some techniques. To overcome this problem, they proposed Real or Fake Text (RoFT) tools to improve the human detector performance. However, as the power of GPT improved, Clark et al. [36] concluded that untrained human detectors could only differentiate between GPT3-authored and human-authored text at random chance levels. The problem of detecting AI-generated texts is further exacerbated by the numerous models that have been developed to generate texts and the ongoing improvement in the quality of the output of the latest models.

B. AUTOMATIC AI-GENERATED TEXT DETECTION

In recent years, numerous language models and systems have been developed to both generate and detect scientific paper content. One notable model is the GLTR proposed by Gehrmann et al. [30], which serves as a fake text detector. The updated version of GLTR leverages three core concepts: word probability, absolute rank based on probability values, and the entropy of the predicted distribution to calculate a likelihood value for each token. These values are then visualized to assist human detectors in judging whether a text is AI-generated. Building on the GLTR model, Kadhimi et al. [37] developed a deep-learning model to automatically classify scientific papers as AI-generated or human-written. The GROVER model, which generates political news articles that are difficult to detect, was countered by Zellers et al. [21] with a defence model that achieved 92% accuracy against both GROVER [38] and GPT-2 systems [6].

Additional language models, such as fastText [39], RoBERTa [40], and BERT [4], have also been harnessed for AI-generated content detection. Solaiman et al. [19] demonstrated that RoBERTa outperformed GPT-2 in the

detection of AI-generated texts. Despite being primarily a language generator, RoBERTa showed higher efficiency as a detector compared to other models [40], [41]. However, its need for large data sets remains a significant drawback. RoBERTa also struggled with content generated by more advanced models, such as those proposed by Wolf et al. [42]. Pegoraro et al. [43] recently declared that current methods fail to accurately detect text generated by ChatGPT-4. Research by Bakhtin et al. [33] explored updated fake text detection using quantitative system results, while Dugan et al. [34] utilized fine-tuned BERT models to address the impacts of text excerpt length and sampling strategy. Varshney et al. [44] introduced formal hypothesis testing with error exponent limits based on cross-entropy and perplexity to tackle challenges from GPT-2 and later models. Maronikolakis et al. [45] achieved 85.7% accuracy with a transformer-based model using multiple classification techniques to detect AI-generated content.

Other approaches include the use of word2vec and Term Frequency-Inverse Document Frequency (TF-IDF) with deep learning for fake content detection by Vijayaraghavan et al. [46] and Jahwahr et al. [47] who focused on styles and discourse models to distinguish AI-generated text from human-written text. Conversely, Bhat et al. [48] highlighted the limitations of discourse and style-based models in this context. Perez's model [49] emphasized the importance of syntactic and semantic textual features for detection purposes.

Some researchers [50], [51] have concentrated on the references section for detecting AI-generated scientific papers. Amancio et al. [52] proposed a topological system, while Nguyen et al. [31] introduced the SciDetect system, which uses an intertextual distance formula to classify content based on its textual features. Their methods involved segmenting long paragraphs and ignoring paragraphs containing fewer than 1000 characters to improve detection accuracy. Cabanac et al. [15] employed rule-based POS tagging in conjunction with a search engine to discover fake scientific papers, though their method was limited to grammar-based detection. The necessity for a corpus of AI-generated text for accurate detection is echoed by Xiong and Huang [50]. Additionally, SCIgen has been used to generate nonsensical computer science research articles for classification purposes. Recently, the attention mechanism has proven its efficiency across various fields [53], [54], [55], [56], [57], with many researchers now combining it with Bidirectional LSTM methods for improved performance in detecting AI-generated text [58], [59], [60].

III. DATASET

To the best of our knowledge, there are few publicly available corpora suitable for use as benchmark datasets to conduct experiments on the detection of AI-generated academic content. However, a benchmark dataset is a prerequisite for any research in the field of generated text detection. Two publicly available datasets[1] were discovered: (1) the Kaggle DAGPap22 dataset and (1) the Liyanage Benchmark dataset. Each of these datasets is described below.

A. KAGGLE DAGPAP22 DATASET

This dataset was placed online on Kaggle for the shared task, namely Detecting Automatically Generated scientific Papers (DAGPap22), of the third workshop on scholarly document processing, a workshop held at the 29th International Conference on Computational Linguistics (COLING 2022). To protect the research from misleading and damaging the scientific community, they created the dataset with a series of concerns [61]. Each scientific paper is labelled with a binary classification: AI-generated or human-written. In the dataset from the AI-generated paper, 5000 excerpts were collected based on Cabanac et al.'s work [17]. They also provided an accessible fivefold larger human-written corpus and AI-generated paper from the common scientific domains and the same documents. In the dataset from the scientific paper, there is a text excerpt which indicates whether the content is AI-generated or human-written. This dataset comes from retracted Scopus and published papers, and in total, 5327 papers are available in the training dataset and 21310 papers for the testing dataset records. There are two columns in each record in which the text or paper content is contained in the first column, and the value of fake or real is provided in the second column, with 1 standing for the generated content and 0 for the human-created texts. The dataset used in this study was obtained from the following source: https://www.kaggle.com/competitions/detecting-generated-scientific-papers/overview.

B. LIYANAGE BENCHMARK DATASET

The second publicly-available dataset for the AI-generated scientific paper detection for research was compiled by Liyanage [32], using a mix of natural human-written text and AI-generated text. The AI-generated texts were designed to be difficult to detect and masquerade as original content to an uninformed reader. They included two types of corpora. One corpus consisted of a hybrid dataset that included original human-written scientific paper content in which AI-generated sentences replaced some sentences. The second corpus consisted of only AI-generated scientific papers. Each of the corpora was collected by considering the situation in which an author created a full text of the paper for submission to a journal. The length of both individual sentences and various sections is not fixed. The mean length of the hybrid dataset was 177 words, while the AI-generated dataset mean was 1247 words. The papers include abstracts, introductions, literature reviews, the proposed approaches and future works. Because of the presence of figures and tables in full papers, they eliminated some sections of the paper before constructing the dataset. In addition, they also

[1]Neither of these datasets is named online, and so we have taken the liberty of naming them.

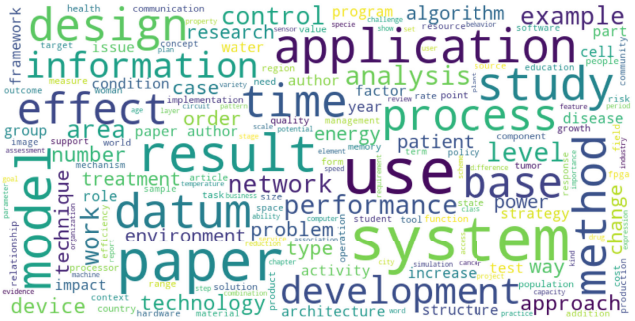

FIGURE 1. Word cloud of nouns in the Kaggle DAGPap22 dataset.

FIGURE 2. Word cloud of the nouns in the Liyanage Benchmark dataset.

considered a situation where authors are not malicious but use AI-generated text to complete certain parts of their papers. There are more than 5000 records in the dataset. As with the Kaggle DAGPap22 dataset, each record contains two columns, one column containing the text of the paper and the other column containing the label of the paper, namely 1 standing for the AI-generated content and 0 standing for human-written content.

Figure 1 shows a word cloud for the 40 most frequent nouns in the Kaggle DAGPap22 dataset. Moreover, the Liyanage Benchmark dataset also has a graph similar to this. Figure 2 shows a word cloud for the 40 most frequent nouns for the Liyanage Benchmark dataset.

Figure 3 shows the frequency of the top 40 word tokens in the Kaggle DAGPap22 benchmark dataset, while the Liyange Benchmark dataset frequency profile is shown in Figure 4. The charts show some shared high-frequency lexical items, such as *use, model* and *performance*.

IV. PROPOSED MODEL

In the proposed model, rule-based models were first applied to extract POS tagging. Based on the POS tagging, a deep learning model built with the Keras library was employed specifically for sequence classification tasks. The workflow is given in Figure 5. A different variant of a bidirectional LSTM model with an attention mechanism added on top was included. Our proposed model draws on the literature review by combining (1) rule-based POS tagging [15], (2) initial module, (3) an attention mechanism [53], [54], [55], [56], [57], (4) a Bidirectional LSTM method [58], [59], [60], (5) feature concatenation, and (6) classification.

The input to the model is a sequence of integers (of length `maxlen`) representing tokens in a text. These integers are first converted into low-dimensional dense vectors using an embedding layer. The embedded sequence is then passed through the initial module and fed into a bidirectional LSTM layer with 64 units, which allows the model to learn from the sequence in both forward and backward directions.

The final hidden state of the bidirectional LSTM layer is fed into the attention mechanism that operates on the output of the bidirectional LSTM layer. The attention mechanism assigns weights to each time step of the output sequence based on its relevance to the final classification task.

Then, the output of the attention mechanism was concatenated with the initial module output. The concatenated output is then fed into the classification module; in the classification module, features are passed through a dense layer with a sigmoid activation function to obtain a binary classification prediction. The attention class defines the attention mechanism used in the model. It consists of three dense layers: two with the same number of units as specified by the input argument ''units'' and one with a single output unit. The method first applies a tanh activation function to a sum of the two dense layers applied to their respective inputs [59], [62]. This produces a score for each time step in the output sequence. The scores are then passed through a softmax function to produce a weight for each time step, representing the relevance of that time step to the final classification task. These weights are then used to compute a context vector, which is a weighted sum of the output sequence. Finally, the context vector and attention weights are returned as outputs and details as visualized in Figure 6.

In summary, in the first stage part of speech (POS) tags were extracted. The maximum length of 5000 was selected based on the sentence length distribution of the text dataset included. The next step was the application of a feature-based hybrid deep learning network that utilizes part-of-speech tagging and combines Bidirectional Long Short-Term Memory (Bi-LSTM) networks with Attention modules. In the Bi-LSTM layer on both sides, where ours selected the output vectors based on the sentence length, it performed better compared to the Bidirectional GRU and others. The initial module extracts local contextual features through convolutional layers, followed by Bi-LSTM layers that capture long-term dependencies in both forward and backward directions. The attention mechanism emphasizes essential sequence elements, improving the model's focus on relevant data. The outputs from the attention and initial modules are concatenated via a residual connection for comprehensive feature representation, which is then passed to dense layers for final classification. This approach strikes a balance between feature richness and computational efficiency.

A. PREPROCESSING

To preprocess the data, the built-in `pos_tag` function of the Natural Language Toolkit (NLTK) library [63] (version

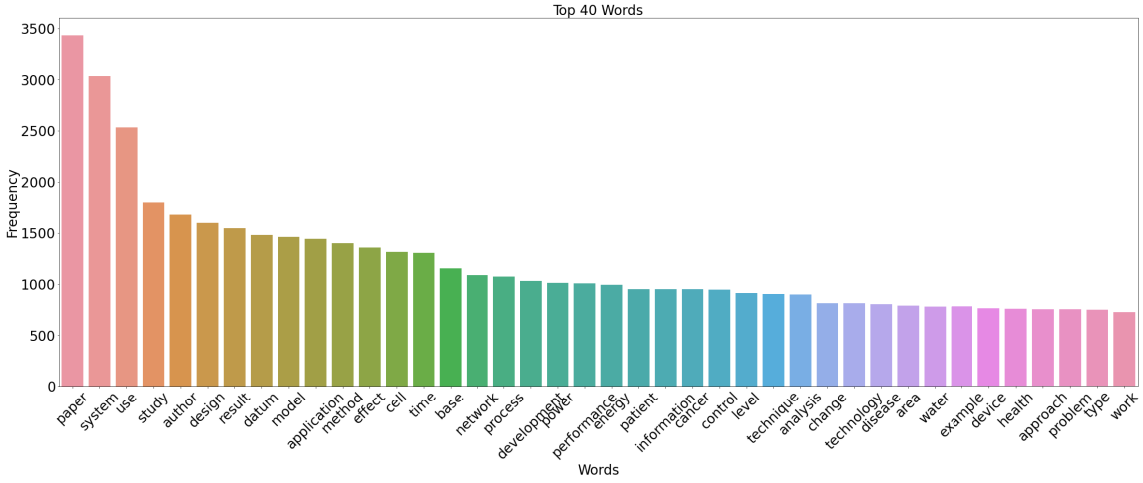

FIGURE 3. Frequency of top 40 words in the Kaggle DAGPap22 dataset.

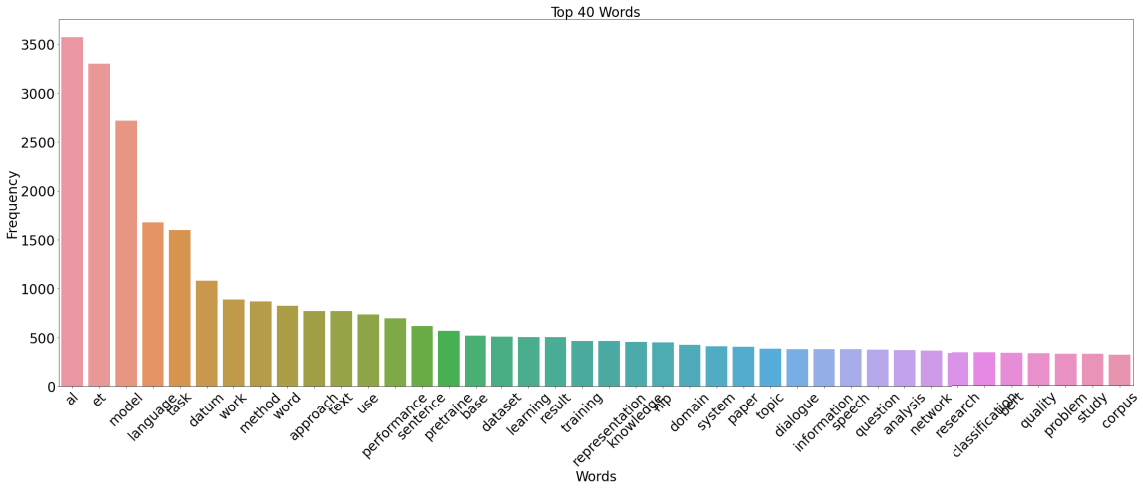

FIGURE 4. Frequency of top 40 words in the Liyanage Benchmark dataset.

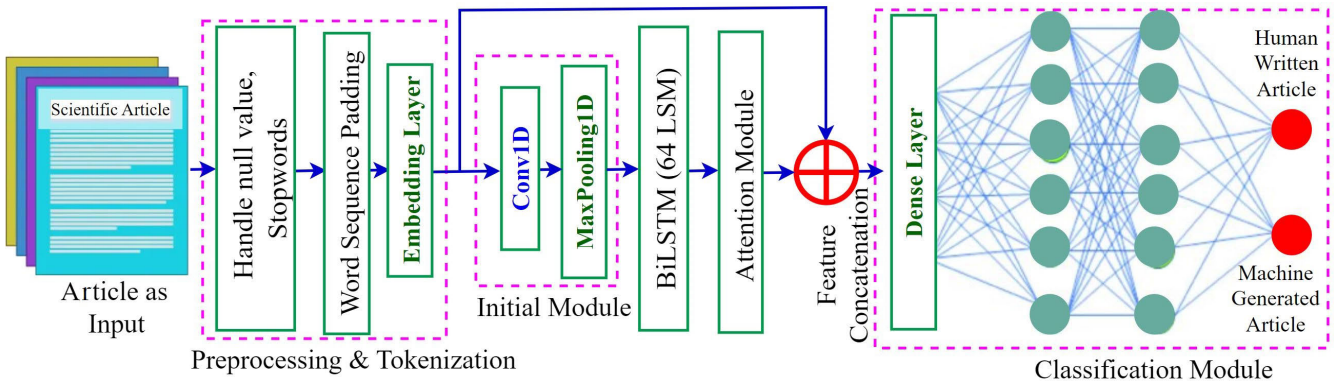

FIGURE 5. Working procedure of the proposed method.

3.8.1) was used for rule-based POS tagging. This function applies a tagging algorithm based on the Penn Treebank tagset [64], which assigns each word token a grammatical POS label (e.g., noun `NN`, verb `VB`, adjective `JJ`, adverb `RB`) by leveraging lexical and contextual rules. First, each sentence was tokenized, splitting it into individual word

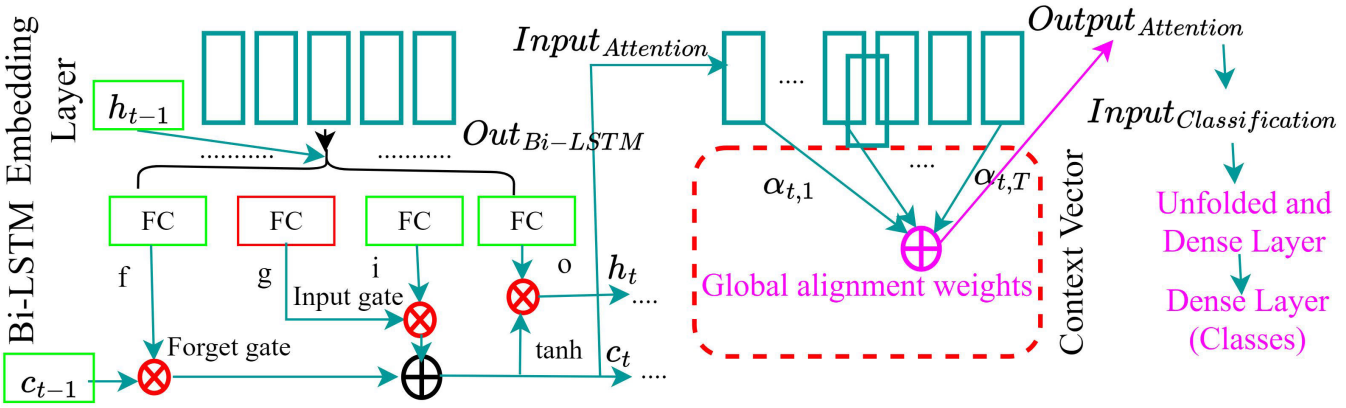

FIGURE 6. Details of the Bi-LSTM and attention layer.

tokens. POS tagging was then performed on these tokens to assign a grammatical POS label to each word. This tagging was conducted on each sentence separately, after which lists for all sentences in both the training and testing sets were compiled. Next, a set of all unique tokens in the tagged sentences was created, which included both word tokens and punctuation marks. Using this set, a dictionary to map each unique word to a numerical index was created, starting at 1, with 0 reserved for padding. For each POS tag, a numerical index was assigned to facilitate its use in the model, focusing on the tags for nouns `NN`, verbs `VB`, adjectives `JJ`, and adverbs `RB`, resulting in the following mapping: '`NN`': 1, '`VB`': 2, '`JJ`': 3, '`RB`': 4. Finally, the maximum length of each text was standardized to 5000 tokens. Sentences longer than this were truncated, while shorter sentences were padded to ensure uniform sequence lengths across all inputs.

B. INITIAL MODULE (EMBEDDING AND CNN WITH MAXPOOLING)

The initial module consists of an embedding layer followed by a 1D convolutional layer and max-pooling. The embedding layer maps input words to dense vector representations, which capture semantic meaning. The Conv1D layer extracts local features from these embeddings, effectively identifying n-gram patterns within the data. MaxPooling reduces the dimensionality of the feature maps and helps retain the most significant information while reducing computational costs. The main reason and novelty of this model is that this initial module ensures that the model efficiently captures local dependencies in the input data, which are essential for understanding the context of word sequences. This approach enhances feature extraction capabilities, improves generalization, and reduces overfitting by condensing the feature space. Suppose the input sequence x is first passed through an embedding layer, followed by a convolutional and max-pooling operation:

$$\mathbf{E} = \text{Embedding}(x) \tag{1}$$

$$\mathbf{F}_{\text{Initial}} = \text{MaxPooling}(\text{ReLU}(\text{Conv1D}(\mathbf{E}))) \tag{2}$$

The output of the initial module fed into the Bidirectional LSTM module.

C. BIDIRECTIONAL LSTM

In this study, we proposed using a Bidirectional LSTM because LSTMs have achieved excellent performance on text classification tasks. Treating an LSTM as a cell allows it to be conceptualized as a closed box, simplifying understanding without requiring an in-depth explanation [60]. This closed box consists of an input and output module and processes the t-th word of a sentence at time step t. After that, it utilizes some internal processing to extract various features, including long-range dependencies, using a number of hidden states, which can be written as h_t and c_t (see Fig. 6). In the RNN, they used only one hidden state h_t, and that can not extract the long-range dependencies feature. The output of LSTM is stacked and applied as input vectors for the forward and backward layer, which makes it a Bidirectional LSTM [58]. The Bi-LSTM processes the output of the initial module $\mathbf{F}_{\text{Initial}}$ to capture sequential dependencies:

$$\mathbf{F}_{\text{Bi-LSTM}} = \text{Bi-LSTM}(\mathbf{F}_{\text{Initial}}) \tag{3}$$

D. ATTENTION MODULE

As previously mentioned, RNN is unable to extract long-range dependencies because of its internal mechanism, but LSTM can extract these features effectively. However, it may occasionally be challenging for LSTM cells to extract long-term dependencies in features because of the non-uniform (i.e., variable) length of the input. The performance of the LSTM may fluctuate if the length of the sentence increases above 30. To overcome the challenges of the LSTM model, an attention mechanism was utilized, which is also capable of extracting long-range dependencies, while offering some feature selection functionality. Rather than concentrating on the entire sequence, the attention mechanism selectively emphasizes specific segments within the arbitrary-length input sequence. Consider a text processing task where h_t represents the number which is passed to the

attention mechanism as input attention (see Fig. 6); attention will not consider all heads coming from the previous layer. It chooses efficient vectors among all input vectors using the probability calculation, which are known as attention weight values. After that, the attention module calculates the context vector by summing the product of input and weight values. It actually calculates the global attention and then employs general scoring criteria to calculate the attention score (see Eq. 4). Scoring criteria mainly indicate how many vectors among all input vectors should be focused on or highlighted. After that, the score is converted into attention weights using the softmax function on scores (see Eq. 5). Finally, the head h_t concatenated with context vector softmax function tanh is applied to obtain the output [58], [59]. The attention mechanism computes a context vector by weighting the Bi-LSTM outputs.

$$AttentionScore(h_t, \bar{h}_s) = \begin{cases} h_t^T \bar{h}_s & \text{dot} \\ h_t^T W_\alpha \bar{h}_s & \text{general} \\ v_\alpha^T tanh\left(W_\alpha [h_t; \bar{h}_s]\right) & \text{concat} \end{cases} \tag{4}$$

$$F_{Att} = \alpha_s(s) = align(h_t, \bar{h}_s) = \frac{exp\left(score(h_t, \bar{h}_s)\right)}{\sum_{\bar{s}} exp\left(score(h_t, \bar{h}_{\bar{s}})\right)} \tag{5}$$

E. FEATURE CONCATENATION AND CLASSIFICATION

The output of the attention module is concatenated with the flattened production from the initial CNN module, forming a residual connection that integrates the original feature representations with those enhanced by the attention mechanism. This approach ensures that the network retains access to high-level and detailed features, effectively preserving valuable information that could be lost through sequential processing alone. By combining local features extracted by the CNN with the long-range dependencies captured by the Bi-LSTM-attention module, the model achieves a richer and more robust representation, enhancing the depth and effectiveness of the overall feature set. The context vector from the attention module is concatenated with the initial module:

$$\mathbf{F}_{\text{Concatenation}} = F_{Initial} \bigoplus F_{Att} \tag{6}$$

The concatenated feature set $\mathbf{F}_{\text{Concatenation}}$ is then passed through a dense layer with a ReLU activation function, followed by a dropout layer to introduce regularization. This prevents overfitting and ensures the model remains resilient during training. The final output layer is a dense layer with a sigmoid activation function designed for binary classification by outputting a probability score for the target class. This combination enables the model to learn complex patterns through the ReLU activation while maintaining generalizability through dropout, resulting in improved performance on unseen data. The combined features are passed through a dense layer with dropout for classification:

$$\mathbf{F}_{\text{dense}} = \text{ReLU}(\mathbf{W}_1 \mathbf{F}_{\text{Concatenation}} + \mathbf{b}_1) \tag{7}$$

$$\hat{y} = \text{Sigmoid}(\mathbf{W}_2 \mathbf{F}_{\text{dense}} + \mathbf{b}_2) \tag{8}$$

where, $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices for the dense layers. $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors. $\hat{y}$ is the final predicted probability for the target class.

V. EXPERIMENTAL EVALUATION

To evaluate the proposed model, two datasets collected from GitHub and Kaggle were harnessed, namely the Liyanage Benchmark dataset and the Kaggle DAGPap22 dataset.

A. ENVIRONMENTAL SETTING

Each dataset was split into 70% for training and 30% for testing. Based on this ratio, for the Kaggle DAGPap22 dataset, the training set comprises 3584 samples and the testing 1766 samples, while the Liyanage Benchmark dataset comprises 205 and 101 samples, respectively. The TensorFlow framework of Python programming [65] was used to implement the experiment in the Google Colab Pro edition environment. There are 25GB GPU in the processing RAM in the environment, known as Tesla P100 [66]. Tensorflow is considered a boon to the deep learning model due to its open-source nature, the concept of the computational graph, and its adaptability and compatibility with minimum resources. The OpenCV Python package was used for the initial image processing task [67]. Various graphs were plotted using the Matplotlib library [67].

B. HYPERPARAMETER TUNING

To optimize the performance and efficiency of our model, key hyperparameters were carefully tuned based on the characteristics of the dataset and the requirements of AI-generated text detection. The maximum sequence length `MAX_LEN` was set to 5000 tokens to ensure sufficient context coverage without excessive computational complexity. The embedding dimension was set to 32, providing compact word representations while maintaining generalization. The attention mechanism, which uses 64 dense units, was incorporated to focus on relevant features within each sequence, enhancing the model's ability to differentiate between human- and AI-generated text. We also included a Conv1D layer with 128 filters and a kernel size of 3, allowing the model to capture local dependencies within text sequences before passing them to bidirectional LSTM layers. In the bidirectional LSTM layers, we used 64 units and applied a dropout rate of 0.3 to mitigate overfitting and improve robustness. We set the recurrent activation function to tanh for stability and used *glorot_uniform* initialization to support efficient gradient flow. Furthermore, the learning rate was set to 0.0001, which provided stable convergence, while gradient clipping with a clip value of 1.0 prevented gradient explosion. Lastly, a dense layer with 64 units and a dropout rate of 0.3 further refined the extracted features before the

TABLE 1. Ablation study of the proposed model.

Ablation Study No	Initial Module (Yes/No)	Number of BiLSTM	No Attention Module	Residual Concatenation (Yes/No)	KAGGLE DAGPAP22 Dataset Accuracy [%]	Trainable Parameters (Millions)	Training Time Sec Per Batch	Required Memory (MB)
1	No	1	1	No	68.23	1.16	1.04	4.43
2	No	2	1	No	80.00	1.26	1.08	4.81
3	Yes	1	1	No	78.23	1.2	0.63	4.90
4	Yes	0	1	No	81.82	1.1	0.08	4.24
5	Yes	2	1	Yes	84.60	41.30	0.26	157.55
Proposed Model	Yes	1	1	Yes	85.00	41.20	0.23	157.17

TABLE 2. State of the art comparison of the proposed model.

Preprocessing	Feature Reduction	Classifier	DAGPap22 Performance [%]	Liyanage Performance [%]
TF-IDF	Random Forest	ETC	82.00	62.00
TF-IDF	Random Forest	LightGBM	79.00	60.00
TF-IDF	PCA	SVM	78.00	60.00
KeyPhrase	No	ETC	79.00	63.00
TF-IDF		SGD	82.00	60.00
TF-IDF	No	LR	81.70	59.00
POS	No	ETC	83.00	60.00
POS	No	LightGBM	80.00	60.00
POS	No	SVM	80.00	60.00
POS	No	ETC	79.00	60.00
POS	No	SGD	81.00	62.00
POS	No	ANN	82.00	62.00
POS		Proposed Method	**85.00**	**88.00**

final output layer, which uses sigmoid activation for binary classification.

These hyperparameters were chosen after extensive experimentation to balance accuracy, computational cost, and model stability, enabling our approach to achieve high accuracy with reduced training and inference times. The training is performed for each dataset for 1000 epochs while 0.000005 was used as the initial learning rate on account of higher fluctuation during the Adam optimizer with Nesterov momentum [68], [69]. Various parameter tuning operations were used for the learning rate and optimizer for the two classes of the study.

C. ABLATION STUDY

An ablation study was conducted to analyze the effects of various components of the proposed model by systematically altering or omitting them and recording the resulting performance. Table 1 summarizes the results of five different model configurations tested on the Kaggle DAGPap22 dataset, including the required parameters, training time, and memory usage for each configuration.

In Study No. 1, the model without the initial module and with a single Bi-LSTM layer achieved an accuracy of 68.23%, with 1.16 million trainable parameters, a training time of 1.04 seconds per batch, and a memory requirement of 4.43 MB. Study No. 2, which incorporated two Bi-LSTM layers without the initial module, showed an improvement in accuracy to 80.00%, accompanied by 1.26 million trainable parameters, a training time of 1.08 seconds per batch, and a memory usage of 4.81 MB. Study No. 3 included the initial module but lacked residual concatenation, resulting in an accuracy of 78.23%, with 1.2 million parameters, a reduced training time of 0.63 seconds per batch, and a memory requirement of 4.90 MB. Study No. 4, which removed Bi-LSTM layers entirely, reached an accuracy of 81.82%, with 1.1 million parameters, the lowest training time of 0.08 seconds per batch, and 4.24 MB of memory. Study No. 5 combined the initial module, two Bi-LSTM layers, and a residual connection, achieving an accuracy of 84.60%. This configuration, however, had a significant increase in trainable parameters at 41.30 million, a training time of 0.26 seconds per batch, and a memory requirement of 157.55 MB.

The complete proposed model, integrating the initial module, a single Bi-LSTM, an attention mechanism, and a residual connection, achieved the highest accuracy at 85.00 %. It maintained 41.20 million trainable parameters, a training time of 0.2325 seconds per batch, and a memory usage of 157.17 MB. This demonstrates that incorporating attention, residual connections, and Bi-LSTM optimizes the model's performance, highlighting the importance of these components in achieving the best results.

D. PERFORMANCE RESULT

The evaluation metric used to evaluate the accuracy, precision, recall and F1 score of the proposed model as an identification model is calculated using Equation 9, 10,11,12

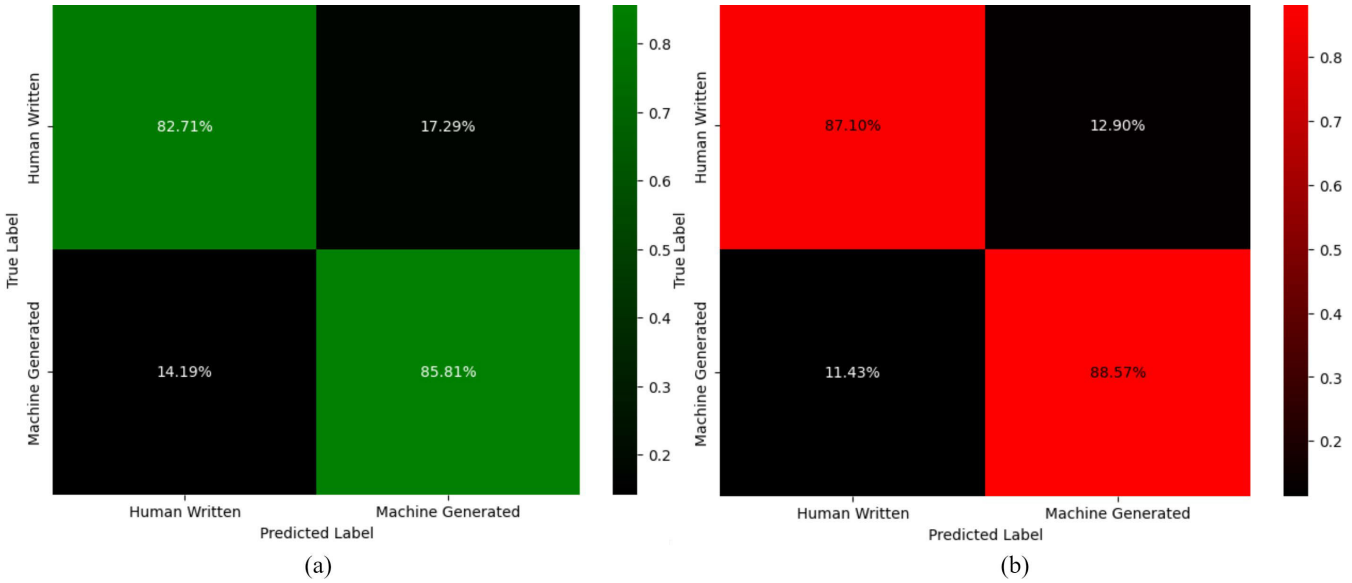

FIGURE 7. Confusion matrix (a) Kaggle DAGPap22 dataset (b) Liyanage Benchmark dataset.

in which TN denotes true negatives, TP denotes true positives, FN false negatives, and FP denotes false positives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

The proposed model was evaluated with two datasets and the performance accuracy for various models was calculated for each dataset. Table 2 shows the performance accuracy of the proposed model and various other models. Initially, experiments on various machine learning algorithms with different combinations of the module, including TF-IDF and key phrase vectorizer module, were carried out.

We selected TF-IDF as a feature extraction technique due to its effectiveness in highlighting distinctive terms within a document by considering their relative frequency across the dataset. This approach enhances the ability of the model to detect subtle differences indicative of AI-generated content. TF-IDF is widely used in text classification and is particularly beneficial for identifying contextually significant words.

The KeyPhrase module was selected to extract significant phrases that encapsulate a document main themes or topics of a document. Unlike single-word features, key phrases capture high-level concepts, providing valuable contextual information for classification. This enables the model to differentiate human-written from AI-generated documents by emphasizing semantically rich and contextually relevant terms.

We employed different machine learning algorithms based on the POS-tagging and achieved 85.00 % and 88.00% accuracy with artificial neural networks for Kaggle DAGPap22 and the Liyanage Benchmark datasets, respectively.

The precision-recall and F1 Score were also calculated for both datasets. For the evaluation of our proposed model on the Kaggle DAGPap22 dataset, we calculated key performance metrics, including precision, recall, and F1-score. The results demonstrated consistent and balanced performance across these metrics, with a precision of 83.5, a recall of 83.00, and an F1-score of 83.00. These findings indicate that the model maintains high accuracy in correctly identifying positive cases while minimizing false positives and false negatives. In addition, for the Liyanage Benchmark datasets, precision, recall and f1-score produced near 88.00% accuracy, which comes from the two classes human written in most cases 88.00% and AI-generated cases 88.00% or vice versa. The balanced precision and recall suggest that the model effectively captures relevant instances and accurately represents the dataset's characteristics, leading to a robust F1 score. This comprehensive evaluation underscores the reliability and efficiency of the proposed model in handling complex data and achieving stable results. Figure 7 shows the confusion matrix for each of the datasets.

E. DISCUSSION

The performance of our proposed model, evaluated on two benchmark datasets, demonstrates its effectiveness in distinguishing AI-generated texts from human-generated content. Achieving accuracies of 85.00% and 88.00%, the model not only surpasses existing algorithms in accuracy but also does so with reduced computational complexity.

Figure 7 (a) shows the confusion matrix for the Kaggle DAGPap22 dataset. This accuracy matrix summarizes the

performance of a classifier on the Kaggle DAGPap22 dataset, which consists of human-generated scientific articles (Class 0) and AI-generated articles (Class 1). The classifier demonstrates a precision of 73.00% for human-generated articles and 91.00% for AI-generated articles, with recall rates of 83.00% and 86.00%, respectively. The resulting F1-scores are 78.00% for human-generated articles and 89.00% for AI-generated articles. With 561 human-generated articles and 1205 AI-generated articles, the classifier achieves an overall accuracy of 85.00% across 1766 instances. The macro average for precision, recall, and F1-score are 82.00%, 84.00%, and 83.00%, respectively, while the weighted averages for these metrics are consistently around 85.00%. Moreover, the classifier correctly classified 82.71% of human-generated articles and 85.81% of AI-generated articles.

The strengths of this classifier are highlighted by its performance metrics. The high precision for AI-generated articles (91.00%) indicates that the classifier effectively identifies AI-generated content with a low false-positive rate. The balanced recall rates for both classes (83.00% for human-generated and 86.00% for AI-generated articles) show that the classifier is also efficient at minimizing false negatives. The overall accuracy of 85.00% demonstrates the classifier's robustness in handling a large dataset with diverse types of articles. The consistent macro and weighted averages for precision, recall, and F1-score reflect the classifier's balanced performance across different classes. These metrics collectively highlight the classifier's reliability and effectiveness in distinguishing between human-generated and AI-generated scientific articles in the Kaggle DAGPap22 dataset.

Figure 7 (b) shows the confusion matrix for the Liyanage Benchmark dataset. This matrix summarizes the performance of a classifier on the Liyanage Benchmark dataset, which consists of human-generated scientific articles (Class 0) and AI-generated articles (Class 1). The classifier demonstrates high precision (87.00% for Class 0 and 89.00% for Class 1) and recall (87.00% for Class 0 and 89.00% for Class 1), resulting in strong F1-scores (87.00% for Class 0 and 89.00% for Class 1). With 31 human-generated articles and 35 AI-generated articles, the classifier achieves an overall accuracy of 88.00% across 66 instances. The macro and weighted averages for precision, recall, and F1-score are all consistently 88.00%, indicating balanced performance across both classes. Additionally, the classifier correctly classified 87.10% of human-generated articles and 88.57% of AI-generated articles. The strengths of this classifier are evident from its performance metrics. High precision and recall for both classes indicate that the classifier effectively distinguishes between human-generated and AI-generated articles with a low rate of false positives and false negatives. The balanced performance metrics, as reflected in the macro and weighted averages, suggest that the classifier performs consistently well across different types of articles. The high overall accuracy of 88.00% and the close F1-scores for both classes highlight the classifier's robustness and reliability. The specific correct classification rates provide clear evidence of its effectiveness, making it a strong tool for accurately identifying the nature of scientific articles in the Liyanage Benchmark dataset. This performance highlights the potential of the proposed approach to be implemented in real-world applications where resource efficiency is as critical as detection accuracy.

One of the key strengths of this model is its hybrid architecture, which integrates part-of-speech tagging, Bi-LSTM networks, and attention modules. By leveraging convolutional layers to extract local contextual features and Bi-LSTM layers to capture long-term dependencies, the model effectively understands the nuances of text sequences from both past and future contexts. The attention mechanism further enhances this capability by emphasizing critical components of the sequences, allowing the model to focus on the most relevant data for classification. This comprehensive feature extraction process ensures that the model can accurately identify subtle differences between human and AI-generated texts.

The integration of a residual connection to concatenate the outputs from the attention and initial modules plays a vital role in maintaining a rich feature set while preventing the loss of important information. This architecture ensures that the model can handle complex patterns in text data, balancing feature richness with computational efficiency.

The final classification through dense layers provides a robust decision-making process, culminating in the high accuracy observed in the evaluations. The implications of this study are significant, especially in the context of academic publishing. The ability to accurately detect AI-generated texts is vital for maintaining the integrity of research publications. With the rapid advancements in natural language processing, the misuse of AI to generate fraudulent or low-quality content poses a serious threat to the academic community.

Our model offers a robust solution to this problem, providing a tool that can help publishers and researchers ensure the authenticity and quality of published works. Furthermore, the reduced computational complexity of the model makes it accessible for implementation in various settings, from academic institutions to large-scale publishing platforms. This efficiency does not compromise its accuracy, making it a viable option for continuous monitoring and detection of AI-generated texts in real-time applications.

The proposed model represents a significant advancement in the detection of AI-generated texts, combining high accuracy with computational efficiency. Its hybrid architecture and comprehensive feature extraction process enable it to effectively differentiate between human and AI-generated content. This study highlights the importance of developing robust detection mechanisms to safeguard the quality and integrity of academic publishing, and our model stands as a promising solution to meet this pressing need. Future research could further enhance this model by exploring additional

linguistic features and expanding its applicability to other domains where AI-generated content poses a risk.

VI. CONCLUSION

In the study, we introduce a novel algorithm that employs advanced machine learning techniques to effectively detect AI-generated texts, addressing the increasing difficulty in distinguishing between human and AI-generated content due to rapid advancements in natural language processing. The misuse of such texts poses significant threats to the quality and integrity of academic publishing, highlighting the urgent need for reliable detection mechanisms to safeguard information quality and trust.

The proposed feature-based hybrid deep learning model incorporates part-of-speech tagging and integrates Bidirectional Long Short-Term Memory (Bi-LSTM) networks with an attention mechanism. The initial module leverages convolutional layers to extract local contextual features, while the Bi-LSTM layers capture long-term dependencies from both past and future sequences. The attention mechanism further refines the model by emphasizing critical sequence components, enhancing focus on relevant data. Residual concatenation of features from the attention and initial modules ensures a comprehensive representation, which is passed through dense layers for final classification. This structure balances feature richness and computational efficiency. The model's evaluation on two benchmark datasets—Kaggle DAGPap22 and the Liyanage Benchmark—showed it outperformed existing algorithms, achieving high accuracy rates of 85.00% and 88.00%, respectively, with reduced computational complexity.

A key limitation of this model is its reliance on part-of-speech tagging designed specifically for English, which may reduce its effectiveness for other languages. Adapting the tagging protocol is necessary for languages with distinct syntactic structures, while for those with less-developed part-of-speech tagging tools, the model may be less applicable. In addition, although the model performs well on current datasets, further refinement may be required to handle domain-specific texts such as medical or legal documents, where specialized terminology could affect accuracy.

Future work will focus on expanding the model to multiple languages by incorporating language-specific or multilingual embeddings and retraining with appropriate part-of-speech tagging schemes. Testing on multilingual datasets will help refine the approach to accommodate diverse syntactic structures and improve cross-linguistic performance. Future work will also include expanding the dataset pool and further comparisons with real-time systems to reinforce the effectiveness of this approach.

ABBREVIATIONS

ANN	Artificial Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
CFG	Context-Free Grammar
CNN	Convolutional Neural Network
ETC	Extra Trees Classifier
GCN	Graph Convolutional Network
GPT	Generative Pre-Training
GLTR	Giant Language model Test Room
GRU	Gated Recurrent Unit
LLM	Large Language Model
LR	Logistic Regression
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
POS	Part Of Speech
RNN	Recurrent Neural Network
RoBERTa	Robustly optimized BERT approach
RoFT	Real or Fake Text
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency

REFERENCES

[1] F. Ullah, S. M. E. Sepasgozar, and C. Wang, "A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms," *Sustainability*, vol. 10, no. 9, p. 3142, Sep. 2018.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[3] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, San Francisco, CA, USA, 2018.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[5] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," 2019, *arXiv:1901.02860*.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 1–12.

[8] I. M. Wani and S. Arora, "A knee X-ray database for osteoporosis detection," in *Proc. 9th Int. Conf. Rel., Infocom Technol. Optim. (Trends and Future Directions) (ICRITO)*, Sep. 2021, pp. 1–5.

[9] G. Lample and A. Conneau, "Cross-lingual language model pretraining," 2019, *arXiv:1901.07291*.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, *arXiv:1910.13461*.

[11] M. Khalil and E. Er, "Will ChatGPT get you caught? Rethinking of plagiarism detection," 2023, *arXiv:2302.04335*.

[12] T. Batane, "Turning to turnitin to fight plagiarism among university students," *J. Educ. Technol. Soc.*, vol. 13, no. 2, pp. 1–12, Apr. 2010.

[13] T. M. Lillis and M. J. Curry, *Academic Writing in a Global Context: The Politics and Practices of Publishing in English*. Abingdon, U.K.: Routledge, 2010.

[14] C. Stokel-Walker, "ChatGPT listed as author on research papers: Many scientists disapprove," *Nature*, vol. 613, no. 7945, pp. 620–621, Jan. 2023.

[15] G. Cabanac and C. Labbé, "Prevalence of nonsensical algorithmically generated papers in the scientific literature," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 12, pp. 1461–1476, Dec. 2021.

[16] J. Blake, E. Pyshkin, and Š. Pavlík, "Automatic detection and visualization of information structure in English," in *Proc. 6th Int. Conf. Natural Lang. Process. Inf. Retr.*, Dec. 2022, pp. 200–204.

[17] G. Cabanac, C. Labbé, and A. Magazinov, "Tortured phrases: A dubious writing style emerging in science. Evidence of critical issues affecting established journals," 2021, *arXiv:2107.06751*.

[18] C. Labbé, D. Labbé, and F. Portet, "Detection of computer-generated papers in scientific literature," in *Creativity and Universality in Language*. Cham, Switzerland: Springer, Jan. 2016, pp. 123–141.

[19] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. Wook Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, and J. Wang, "Release strategies and the social impacts of language models," 2019, *arXiv:1908.09203*.

[20] Madhumita Murgia, "ChatGPT maker OpenAI unveils new model GPT-4," Financial Times, London, U.K., Mar. 15, 2023.

[21] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 1–16.

[22] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.

[23] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection," in *Proc. 34th Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2020, pp. 1341–1354.

[24] R. Faris, H. Roberts, B. Etling, N. Bourassa, E. Zuckerman, and Y. Benkler, "Partisanship, propaganda, and disinformation: Online media and the 2016 U.S. presidential election," *Berkman Klein Center Res. Publication*, vol. 2017, pp. 1–142, Aug. 2017.

[25] C. Wardle and H. Derakhshan, "Information disorder: Toward an interdisciplinary framework for research and policy making," Council Eur., Strasbourg, France, Tech. Rep. First Draft, 2017.

[26] Q. Wang, L. Huang, Z. Jiang, K. Knight, H. Ji, M. Bansal, and Y. Luan, "PaperRobot: Incremental draft generation of scientific ideas," 2019, *arXiv:1905.07870*.

[27] Q. Wang, Q. Zeng, L. Huang, K. Knight, H. Ji, and N. Fatema Rajani, "ReviewRobot: Explainable paper review generation based on knowledge synthesis," 2020, *arXiv:2010.06119*.

[28] E. S. Abd-Elaal, S. H. Gamage, and J. E. Mills, "Artificial intelligence is a tool for cheating academic integrity," in *Proc. 30th Annu. Conf. Australas. Assoc. Eng. Educ.* Brisbane, QLD, Australia: Engineers Australia Brisbane, 2019, pp. 397–403.

[29] M. Bao, J. Li, J. Zhang, H. Peng, and X. Liu, "Learning semantic coherence for machine generated spam text detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[30] S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical detection and visualization of generated text," 2019, *arXiv:1906.04043*.

[31] M. T. Nguyen and C. Labbé, "Engineering a tool to detect automatically generated papers," in *Proc. BIR Bibliometric-enhanced Inf. Retr.*, Mar. 2016, pp. 1–12.

[32] V. Liyanage, D. Buscaldi, and A. Nazarenko, "A benchmark corpus for the detection of automatically generated text in academic publications," 2022, *arXiv:2202.02013*.

[33] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, and A. Szlam, "Real or fake? Learning to discriminate machine from human generated text," 2019, *arXiv:1906.03351*.

[34] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, "Automatic detection of generated text is easiest when humans are fooled," 2019, *arXiv:1911.00650*.

[35] L. Dugan, D. Ippolito, A. Kirubarajan, and C. Callison-Burch, "RoFT: A tool for evaluating human detection of machine-generated text," 2020, *arXiv:2010.03070*.

[36] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith, "All That's 'Human' is not gold: Evaluating human evaluation of generated text," 2021, *arXiv:2107.00061*.

[37] S. Al-Kadhimi and P. Löwenström, "Identification of machine-generated reviews: 1D CNN applied on the GPT-2 neural language model," M.S. thesis, KTH Roy. Inst. Technol., 2020.

[38] R. Gagiano, M. M.-H. Kim, X. J. Zhang, and J. Biggs, "Robustness analysis of Grover for machine-generated news detection," in *Proc. 19th Annu. Workshop Australas. Lang. Technol. Assoc.*, 2021, pp. 119–127.

[39] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[40] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[41] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "TweepFake: About detecting deepfake tweets," *PLoS ONE*, vol. 16, no. 5, May 2021, Art. no. e0251415.

[42] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstration*, 2020, pp. 38–45.

[43] A. Pegoraro, K. Kumari, H. Fereidooni, and A.-R. Sadeghi, "To ChatGPT, or not to ChatGPT: That is the question!" 2023, *arXiv:2304.01487*.

[44] L. R. Varshney, N. Shirish Keskar, and R. Socher, "Limits of detecting text generated by large-scale language models," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2020, pp. 1–5.

[45] A. Maronikolakis, H. Schutze, and M. Stevenson, "Identifying automatically generated headlines using transformers," 2020, *arXiv:2009.13375*.

[46] S. Vijayaraghavan, Y. Wang, Z. Guo, J. Voong, W. Xu, A. Nasseri, J. Cai, L. Li, K. Vuong, and E. Wadhwa, "Fake news detection with different models," 2020, *arXiv:2003.04978*.

[47] G. Jawahar, "Detecting human written text from machine generated text by modeling discourse coherence," Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, Final Project, 2020.

[48] M. M. Bhat and S. Parthasarathy, "How effectively can machines defend against machine-generated fake news? An empirical study," in *Proc. 1st Workshop Insights Negative Results NLP*, 2020, pp. 48–53.

[49] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," 2017, *arXiv:1708.07104*.

[50] J. Xiong and T. Huang, "An effective method to identify machine automatically generated paper," in *Proc. Pacific–Asia Conf. Knowl. Eng. Softw. Eng.*, Dec. 2009, pp. 101–102.

[51] A. Lavoie and M. Krishnamoorthy, "Algorithmic detection of computer generated text," 2010, *arXiv:1008.0706*.

[52] D. R. Amancio, "Comparing the topological properties of real and artificially generated scientific manuscripts," *Scientometrics*, vol. 105, no. 3, pp. 1763–1779, Dec. 2015.

[53] A. S. M. Miah, Md. A. M. Hasan, and J. Shin, "Dynamic hand gesture recognition using multi-branch attention based graph and general deep learning model," *IEEE Access*, vol. 11, pp. 4703–4716, 2023.

[54] A. S. M. Miah, J. Shin, M. A. M. Hasan, and M. A. Rahim, "BenSignNet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network," *Appl. Sci.*, vol. 12, no. 8, p. 3933, Apr. 2022.

[55] A. S. M. Miah, M. A. M. Hasan, J. Shin, Y. Okuyama, and Y. Tomioka, "Multistage spatial attention-based neural network for hand gesture recognition," *Computers*, vol. 12, no. 1, p. 13, Jan. 2023.

[56] A. Saleh Musa Miah, J. Shin, M. Al Mehedi Hasan, M. Abdur Rahim, and Y. Okuyama, "Rotation, translation and scale invariant sign word recognition using deep learning," *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2521–2536, 2023.

[57] J. Shin, A. S. Musa Miah, M. A. M. Hasan, K. Hirooka, K. Suzuki, H.-S. Lee, and S.-W. Jang, "Korean sign language recognition using transformer-based deep neural network," *Appl. Sci.*, vol. 13, no. 5, p. 3029, Feb. 2023.

[58] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[59] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," 2016, *arXiv:1611.01603*.

[60] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[61] H. Else, "'Tortured phrases' give away fabricated research papers," *Nature*, vol. 596, no. 7872, pp. 328–329, Aug. 2021.

[62] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling radiological language with bidirectional long short-term memory networks," 2016, *arXiv:1609.08409*.

[63] S. Bird, "NLTK: The natural language toolkit," in *Proc. COLING/ACL Interact. Presentation Sessions*, 2006, pp. 69–72.

[64] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," *Comput. linguistics*, vol. 19, no. 2, pp. 313–330, Apr. 1993.

[65] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[66] K. Tock, "Google CoLaboratory as a platform for Python coding with students," *RTSRE Proc.*, vol. 2, no. 1, pp. 1–13, Dec. 2019.

[67] S. Gollapudi, "OpenCV with Python," in *Learn Computer Vision Using OpenCV*. Cham, Switzerland: Springer, 2019, pp. 31–50.
[68] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256.
[69] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. Workshop Given Int. Conf. Learn. Represent.*, Feb. 2016, pp. 1–17.

JOHN BLAKE (Member, IEEE) received the M.Sc. degree in computer science, the M.B.A. degree, the M.Ed. degree in applied linguistics, the M.A. degree in creative writing, the M.A. degree in mathematics education, and the Ph.D. degree in applied linguistics from the University of Aston, Birmingham, U.K. He is a Chartered IT Professional and a Professional Member of the British Computer Society. Currently, he is a Professor with the School of Computer Science and Engineering, The University of Aizu, Japan. He has authored or co-authored more than 80 articles. His research primarily utilizes corpus linguistics to analyze texts and computational linguistics to develop pattern-searching tools and pipelines. He also serves as an Academic Editor for *PLOS One* and reviews for multiple journals in the fields of natural language processing, linguistics, and education.

ABU SALEH MUSA MIAH (Member, IEEE) received the B.Sc.Engg. and M.Sc.Engg. degrees (Hons.) in computer science and engineering from the Department of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh, in 2014 and 2015, respectively, and the Ph.D. degree in computer science and engineering from The University of Aizu, Japan, in 2024, under a Scholarship from the Japanese government (MEXT). He assumed the positions of a Lecturer and an Assistant Professor with the Department of Computer Science and Engineering, Bangladesh Army University of Science and Technology (BAUST), Saidpur, Bangladesh, in 2018 and 2021, respectively. He has been a Visiting Researcher (a Postdoctoral Fellow) with The University of Aizu, since April 2024. He has authored and co-authored more than 50 publications in widely cited journals and conferences. His research interests include AI, ML, DL, human activity recognition (HCR), hand gesture recognition (HGR), movement disorder detection, parkinson's disease (PD), HCI, BCI, and neurological disorder detection.

KRZYSZTOF KREDENS received the M.A. degree in English studies and the Ph.D. degree in English linguistics from the University of Lodz. He is currently a Reader in forensic linguistics and the Director of the Centre for Forensic Text Analysis, Aston University. He has a variety of research outputs in the field and ample casework experience, both as an expert witness and in policing contexts. He is registered as an expert in forensic linguistics on the National Crime Agency's Expert Advisers Database. He is interested in the theoretical underpinnings of the notion of idiolect, but he also works on more practical problems related to linguistically enabled offender identification. He also has an interest in interactionally and institutionally engendered communication barriers in the justice system and how they can be removed. His main research focus is on individual variation in language use with reference to forensic authorship analysis.

JUNGPIL SHIN (Senior Member, IEEE) received the B.Sc. degree in computer science and statistics and the M.Sc. degree in computer science from Pusan National University, South Korea, in 1990 and 1994, respectively, and the Ph.D. degree in computer science and communication engineering from Kyushu University, Japan, in 1999, under a Scholarship from the Japanese Government (MEXT). He was an Associate Professor, a Senior Associate Professor, and a Full Professor with the School of Computer Science and Engineering, The University of Aizu, Japan, in 1999, 2004, and 2019, respectively. He has co-authored more than 350 published papers for widely cited journals and conferences. His research interests include pattern recognition, image processing, computer vision, machine learning, human–computer interaction, non-touch interfaces, human gesture recognition, automatic control, parkinson's disease diagnosis, ADHD diagnosis, user authentication, machine intelligence, bioinformatics, handwriting analysis, recognition, and synthesis. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He served as the program chair and as a program committee member for numerous international conferences. He serves as an Editor for IEEE journals Springer, Sage, Taylor and Francis, *Sensors* (MDPI), *Electronics* (MDPI), and *Tech Science*. He serves as an Editorial Board Member for *Scientific Reports*. He serves as a reviewer for several major IEEE and SCI journals.