scientific reports

Check for updates

OPEN

Using generative adversarial network to improve the accuracy of detecting AI-generated tweets

Yang Hui

This paper provides a novel approach using state-of-the-art generative Artificial Intelligence (AI) models to enhance the accuracy of machine learning methods in detecting AI-generated texts; the underlying generative capabilities are used along with ensemble-based learning methods for the exact characterization of created text attributes. Four basic steps are involved in the proposed methodology. The first step of the text process is the preprocessing stage itself consisting of several steps for the purification of irrelevant data. These stages include noise removal, text tokenization, removal of stop-words, word normalization, and handling uncommon words. In the next step, feature engineering and text representations are done whereby every preprocessed text is represented by a square matrix. This matrix encapsulates data about word correlations, cooccurrence, and word weights. The third step is Generative Adversarial Network (GAN)-based feature extraction, using a GAN model to extract efficient features in classifying the texts based on their creator type. After that, it turns the discriminator part into a strong feature extraction model. The fourth step is weighted Random Forest (RF)-based detection, with the features extracted by the discriminator of GAN serving as input to the RF-based detection model. This approach has covered the differences between texts generated by a human and that generated by Artificial Intelligence, with a significant improvement of 99.60% average accuracy, representing a 1.5% improvement against comparative methods.

Keywords Artificial Intelligence, AI-generated tweets, Generative adversarial network, Random forest, Text analysis

The advent of generative AI has transformed content generation, allowing us to create remarkably genuine and varied outputs, including photos, videos, texts, and music that closely resemble media made by humans. AI-driven technologies have become widespread in all sectors of our society and offer dependable assistance in numerous applications. They can speed the process of writing emails and texts, as well as boost programming by providing powerful code completion capabilities. Despite its remarkable advantages, generative AI also carries the potential for substantial adverse consequences. An urgent issue is the capacity to produce convincingly authentic yet deceptive content, which can be employed to disseminate misleading information, manipulate individuals, and shape public sentiment[1].

AI-generated text detection is a study on artificial intelligence technologies in the recognition and discrimination of genuine text from fabrications, deceptions, and unsuitable information[2]. Deep learning technology has rapidly been diffused, and AI-generated text has been put into wide application. However, this technology has also brought a number of problems, including wrong information distribution and privacy disclosure[3]. Therefore, detecting and identifying text AI has generated is the focus nowadays in artificial intelligence. The history of research into AI-generated text detection goes back to the year when the Natural Language Processing (NLP) field was born. Deep learning technologies have really hugely increased the capabilities of AI-enabled text generation after models such as Recurrent Neural Networks (RNN)[4], Long Short-Term Memory Networks (LSTM)[5], and Transformer[6] were developed. These models can generate high quality text content, be it papers, dialogues, news, or any other. They can easily be used and are readily available for the production of misinformation to mislead consumers or spread harmful content[7]. A critical challenge in this context is the ability to accurately detect AI-generated text, particularly on platforms like Twitter where such content can be misused for malicious purposes. Existing machine learning methods for text classification often struggle to differentiate between human-written and AI-generated text. This motivates the exploration of novel approaches that leverage the capabilities of generative AI models themselves to enhance the accuracy of AI text

School of Humanities and Law, Zhengzhou Shengda University, Zhengzhou 451191, Henan, China. email: huiyang0909@126.com

detection. This study proposes a novel methodology that integrates generative AI with ensemble-based learning techniques to achieve superior performance in identifying AI-generated tweets.

The main innovations of this work include an integration of the GAN-based methods with ensemble-based approaches for the extraction of unique features that help in characterizing AI-generated texts, structured text representation strategies that encode a tremendous amount of word-level features, and the uniquely different transformation of the GAN discriminator into a powerful feature extractor. This is a more holistic approach because it deals with inherent characteristics of generative AI models that impressively boost the ensemble Model Rating in classifying human-written from AI-generated, developing its niche in the area of authorship detection. The contributions of the paper are enumerated below:

- Novel GAN-based Feature Extraction: Our model proposes a new way of feature extraction in which a generative adversarial network is used. By successfully characterizing various features and representations that are difficult to characterize in a conventional way, the GAN is able to distinguish between human written and AI written texts.
- Synergistic Integration of GAN and Ensemble Learning: In the proposed model, feature extraction using GAN is integrated with the weighted ensemble learning system to improve the results of the text detection produced by AI.
- Extensive Text Representation: The suggested method makes use of a thoroughly structured and comprehensive text representation approach that includes a variety of word characteristics and their interdependencies. The subsequent stages of feature extraction and classification have this representation as a solid foundation.

The paper proceeds as follows: section "Related works" examines at similar works. In section "Research methodology", the proposed approach is described; in section "Research finding", the results of its implementation are given; and in section "Conclusion", conclusions are reached.

Related works

In this section, the work done in recent years has been reviewed. Tran et al.[8] provided a Vietnamese text dataset, ViDetect, to approximately establish the adaptability and efficiency of many state-of-the-art AI-generated text recognition methods on the Vietnamese language.

Wang et al.[9] proposed the method of DetectGPT-SC, which further improved the classification between machine-generated and human-written text with the help of ChatGPT. The results confirm that this method works due to self-consistency with masked prediction and show much better results than all previous detectors on all tested tasks.

Bhattacharjee and Liu[10] assessed the performance of ChatGPT regarding detecting AI-generated language by contrast with human-authored text. They, in turn, provided valuable insights into automated detection systems using publicly available datasets.

Ghosal et al.[11] provided a concise survey about AI-generated text detection, its power, and its limitations; further ahead, it went to present an in-depth analysis of some key open problems related to current research in this very area.

In the shared task AuTexTification, Fernando et al.[12] proposed a method for optimizing BERT-based with GPT-2 Small. Thereby, accordingly, it obtained an F1-macro score at the top with GPT-2 Small.

Lokna et al.[13] used was oriented to human-written samples in AI-generated text detection. Using the atomic interpretable grammatical patterns, identification accuracy was significantly improved from 43 to 86% across various domains and language models.

Ghosal et al.[14] offered a succinct summary of AI-generated text detection, exploring its potential and constraints, and delving into a comprehensive analysis of crucial unresolved issues pertaining to ongoing research.

Abburi et al.[15] provided a simple, yet strong approach for the detection of which text is artificial and which is human-written. They applied condensed ensembling, whereby some performance improvements of between 0.5 and 100% over the prior methods were realized.

Soumya et al.[16] contributed a concise survey on the detection of AI-generated text, investigating its potential and constraints, and going further to provide an in-depth exploration of key open challenges in current research.

Akram[17] was interested in developing a wide range of datasets to test the quality of AI content detectors for multidomain materials generated by ChatGPT. The dataset includes papers, abstracts, stories, news items, product reviews with accuracy rates that range from 55.29 to 97.0%.

The work by An[18] is a survey of a great variety of AIGC detection methods. He comments that most of them have low identification accuracy against the latest very large language models, like ChatGPT or GPT-4. Future work is suggested in the line of AIGC quality identification.

Zhang et al.[19] conducted a study on the robustness of the detection methods of AI-generated text against prompt editing and reported that edits to prompts can drop F1 scores by over 50%. This again brings out that continued challenge in detecting AI-generated text with language models' unending ramping up of strength.

Wang et al.[20] proposed a new method of sentence-level AI-generated text detection: Sequence X (Check) GPT, which recommends log probability lists from white-box big language models to outperform previous methods in their generalization ability.

Wang et al.[21] proposed a text identification model based on the Bidirectional Encoder Representations from Transformers (BERT) algorithm, which had enhanced accuracy and stability, while the model also demonstrated great generalizability with wide application prospects in many industries.

Wang et al.[22] presented IDEATE, a hierarchical graph network that exploited both internal and external factual structures in the detection of AI-generated text. IDEATE always performed better than previous methods for

detecting more complicated language models and showed the necessity of external evidence. Table 1 summarizes the studied works.

Research methodology

In this paper, an attempt has been made to enhance the performance of machine learning techniques in identifying texts generated by AI using the capabilities of generative AI. One of the fundamental requirements for achieving this goal is leveraging a rich dataset with the ability to cover a wide spectrum of artificial content and various topics. Therefore, in the following, the specifications of the dataset used to fulfill this requirement are described, followed by the presentation of the details of the proposed method.

Data

The dataset employed in the current research consists of 1,737,000 textual messages covering diverse topics such as politics, events, health, science and technology, sports news, personal texts, and more. Each topic category includes a minimum of 55,000 messages. The text of each message contains a minimum of 48 characters and a maximum of 280 characters, aligning with the technical requirements of tweets on Network X. All samples in this dataset are in English and may include not only alphabets and numbers but also web links and special characters. 837,000 samples belong to texts generated by human users, extracted from the pages of X Users. All personal names and identifying information present in the texts have been replaced with identity-free keywords. On the other hand, 900,000 samples belong to texts generated by generative AI models such as Gemini, GPT-4, Copilot, Claude, and LLaMA. This dataset serves as input for the proposed model in identifying texts created by AI during its training and testing phases.

Proposed model

This section elaborates on the details of the proposed method. In this research, the capabilities of generative AI models are utilized to improve the performance of machine learning techniques in identifying texts created by AI. These capabilities are seamlessly integrated with ensemble-based learning techniques to effectively describe relevant features of artificial texts.

The proposed method aims to achieve the primary objective outlined in this study through the following steps:

1. Text Preprocessing.
2. Feature Engineering and Text Representation.
3. GAN-Based Feature Extraction.
4. RF-Based Detection.

These processes are visualized in a diagram (Fig. 1). According to this figure, the proposed method begins with text preprocessing, which includes partial stages such as noise removal, tokenization, stop-word removal, word normalization, and handling of rare words. By applying these processes, non-informative data, which constitutes a significant portion of the information, is filtered out, significantly narrowing down the problem space for feature representation of textual content. In the second step, each preprocessed text is represented in a new matrix format. In this approach, each tweet is described as a square matrix that simultaneously captures information about word correlations, co-occurrence in the presence of other words, and the weight of words

Reference	Year	Research goal	Method	Limitation
Tran et al.[8]	2024	Evaluate AI-generated text detection methods on Vietnamese text	Dataset creation and Previous Detection Methods	Limited scope to Vietnamese language
Wang et al.[9]	2023	Improve detection of AI-generated text	Self-consistency with masked predictions	Relies on ChatGPT's capabilities
Bhattacharjee and Liu[10]	2024	Assess ChatGPT's ability to detect AI-generated text	Comparative analysis	Limited to ChatGPT's capabilities
Ghosal et al.[11]	2023	Survey AI-generated text detection	Literature review	Does not propose a new method
Aguilar-Canto et al.[12]	2023	Optimize BERT-based models for AI-generated text detection	Model optimization	Limited to BERT-based models
Lokna et al.[13]	2023	Detect AI-generated text using grammatical patterns	Pattern-based approach	Relies on human-written samples
Ghosal et al.[14]	2023	Survey AI-generated text detection	Literature review	Does not propose a new method
Abburi et al.[15]	2023	Detect AI-generated text using ensemble methods	Ensemble learning	Limited to specific ensemble techniques
Soumya et al.[16]	2023	Survey AI-generated text detection	Literature review	Does not propose a new method
Akram[17]	2023	Create a dataset for testing AI-generated text detectors	Dataset creation	Limited to ChatGPT-generated content
An[18]	2023	Survey AI-generated text detection methods	Literature review	Identifies limitations of existing methods
Zhang et al.[19]	2024	Evaluate the robustness of AI-generated text detection against prompt editing	Experimental evaluation	Focuses on prompt editing
Wang et al.[20]	2023	Detect AI-generated text at the sentence level	Sequence X (Check) GPT	Limited to sentence-level detection
Wang et al.[21]	2024	Detect AI-generated text using BERT	BERT-based model	Limited to BERT architecture
Wang et al.[22]	2024	Detect AI-generated text using internal and external factual structures	Hierarchical graph network	Relies on external factual information

Table 1. Table of the related works.

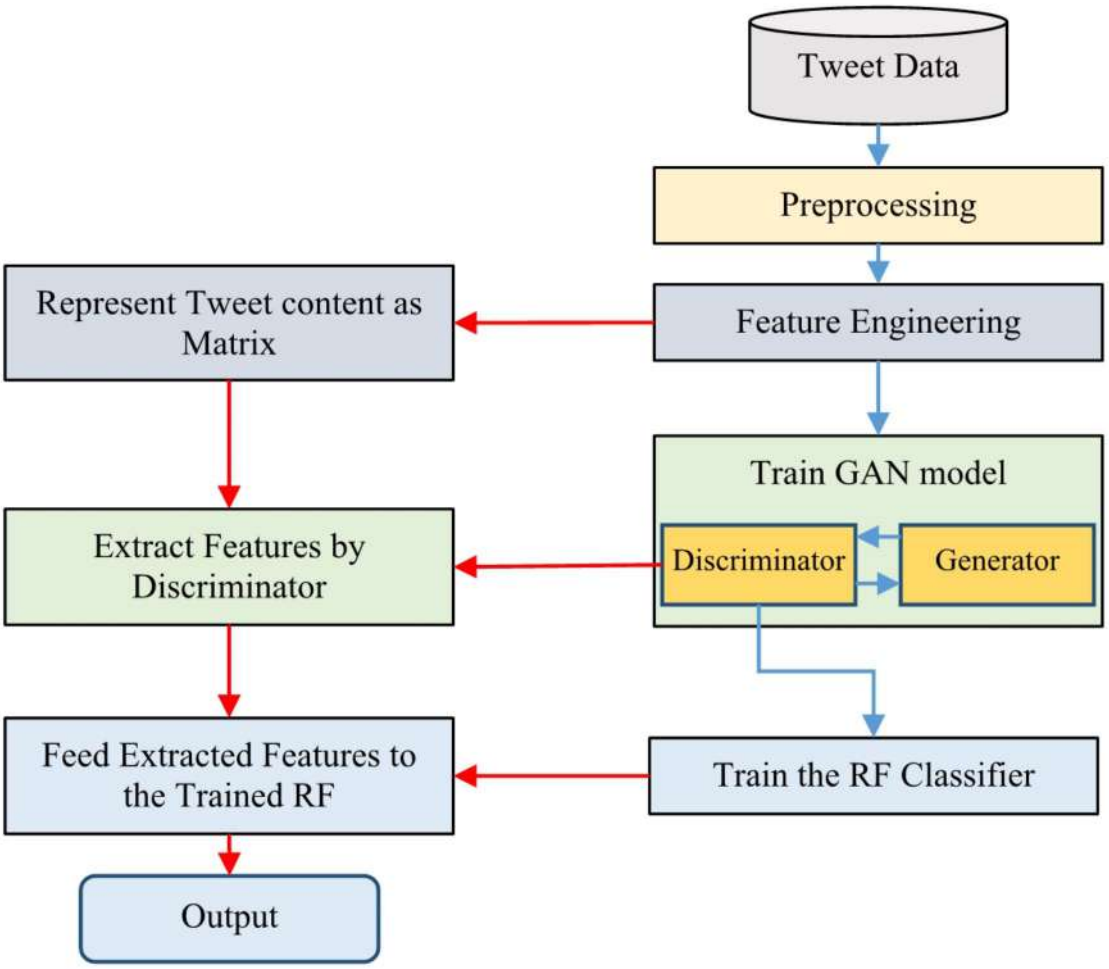

Fig. 1. Proposed method diagram.

constituting the text. The resulting set of matrices describes the entire initial dataset in a structured and coherent format, including features such as simultaneous descriptions, statistical features, and occurrence patterns of words. These features make it ideal for detecting text authors. To achieve this, the third step of the proposed method employs a GAN model. The purpose of introducing this GAN model is solely to obtain an effective feature extraction model for accurately distinguishing texts based on their creator type, Therefore, the generator applications of this model (the generator section in GAN) are ignored. By training the GAN model based on the representation matrices obtained in the second step, the discriminator part of the GAN model transforms into a powerful feature extraction model with the ability to distinguish the source of representation matrices. Therefore, in the final step of the proposed method, the extracted features by the discriminator part of the GAN model are used as input for an RF-based detection model. The RF model employed in the proposed method utilizes a weighted ensemble strategy for detecting the target variable. This technique significantly enhances the ensemble model's capability in more accurate identification of text authors.

Preprocessing

The initial step in the proposed approach involves data preprocessing, a critical phase that lays the foundation for effective model training. During this stage, raw tweet data is transformed into a suitable format for matrix representation and subsequent GAN training. This process comprises several key steps as follows:

a. Data cleaning and noise removal:

Raw tweet data often contain noise and irrelevant information that can adversely affect the performance of the final detection model. Therefore, the preprocessing process begins by cleaning tweets from unrelated elements:

- URLs, numbers, currency units, and hashtags: Although these pieces of information may be informative in some cases, they can divert the model's focus from the main content in applications related to text authorship identification. Therefore, removing them can be effective in constraining the problem space. In the proposed method, each of these cases is replaced with a common keyword that merely describes the presence of the

component in the text. Consequently, URLs, numbers, currency units, and hashtags are respectively replaced with the keywords httpAddr, numIDs, currIDs, and hstIDs.
- User IDs and emojis: Similar to hashtags, mentions (e.g., "@username"), and emojis can be replaced with generic tokens based on the research objective. It should be noted that the focus of the proposed model is not on sentiment analysis; therefore, removing emojis is an effective step in reducing data complexity. Consequently, user IDs and emojis are respectively replaced with corresponding generic tokens.
- Punctuation Marks and Special Characters: Punctuation marks (e.g., comma, point, exclamation point) and all special characters outside the standard alphabet are removed.

For example, consider the following tweet: 'Great news! Our new product launch is a huge success! #innovation #tech #excited https://www.sample-addr.com.' After preprocessing, the resulting tweet is: 'Great news product launch success excited httpAddr hstIDs.'

b. Tokenization and character conversion:

In the second preprocessing step, the cleaned tweets are tokenized into constituent words. This approach allows for a more effective analysis of word relationships within a tweet. Additionally, converting all words to lowercase ensures model stability, which is applied at the end of this stage.

c. Removal of stop words:

Stop words, such as common words like "a," "be," and "is," have little semantic meaning on their own. Removing them can improve the model's focus on content-rich words. In the proposed method, a pre-defined list of English stop words is used to remove this set from the input texts.

d. Text normalization:

Text normalization can involve partial processes such as stemming or lemmatization. In this study, stemming is employed to reduce data complexity. Therefore, in this step, the Porter algorithm is specifically employed. For instance, after removing irrelevant words and normalizing the text, the cleaned tweet results in the following: 'great product launch success excite httpaddr hstids.'

e. Handling rare words:

Rare words refer to words that appear only a few times in the dataset and including them may pose challenges during model training. To address this issue, an empirical threshold based on word frequency in the entire dataset is used to identify rare words. All such words are replaced with the generic label "rareWord.

By following these steps, a preprocessed dataset of tweets is obtained, which is suitable for the second step of the proposed method.

Feature engineering and text representation

In the second step of the proposed approach, after preprocessing the tweets, a set of descriptive features for texts is engineered to more effectively represent word relationships and importance within each tweet. This stage plays a crucial role in translating textual content into a suitable format for GAN and subsequent classification by the RF model. The proposed method utilizes a co-occurrence matrix as a powerful tool for representing word interactions.

The proposed co-occurrence matrix is a square matrix where each row and column describe a unique word present in the preprocessed tweets. The values in this matrix indicate patterns of co-occurrence between these words within the text. The proposed co-occurrence matrix consists of three distinct components to provide a more accurate representation of textual content:

Upper Triangular Component (Word Correlations): This component describes the correlation between pairs of words. The Pearson correlation coefficient, which measures the frequency of word pairs relative to their separate frequencies, is used for this purpose. In this approach, high correlation values indicate a strong relationship between word pairs corresponding to the row and column present in the matrix element.

Lower Triangular Component (Word Co-occurrence): This component indicates the raw co-occurrence count of each word pair in the tweet. For example, for the given tweet: 'Great news! Our new product launch is a huge success!', after preprocessing, the lower triangular component for the intersection of the row and column corresponding to the words 'success' and 'excited' is 1, indicating that this word pair occurs once in this specific tweet.

Main Diagonal (Word Weights): In the proposed matrix structure, the diagonal elements represent the weight of each word in the tweet separately. To achieve this, the TF-IDF measure is used, which considers both the term frequency (frequency within a tweet) and the inverse document frequency (rarity across the entire dataset) of each word. Thus, this measure helps assess the importance of each word in a specific tweet domain and enhances the information richness provided by the representation matrices.

By combining these elements, the proposed co-occurrence matrix provides a rich representation of word interactions and relative importance within each tweet. This information is essential during the feature extraction process for the GAN model.

The proposed co-occurrence matrix structure for representing textual content in tweets offers several advantages:

- Captures word relationships: By explicitly recording the frequency of word occurrences, this matrix goes beyond simple approaches to describe word abundances and provides insights into natural language patterns in real tweets compared to synthetic samples.
- Describes word importance: The TF-IDF weights determine the significance of words based on their frequency within tweets and rarity across the overall dataset, allowing the model to focus on content-rich terms.
- Suitable for GAN training: The proposed matrix format presents a structured representation that GANs can easily process and utilize.

This feature engineering step translates preprocessed tweets into a format that encapsulates the essence of tweet structure and word relationships. The co-occurrence matrix serves as a valuable input for GANs, as further elaborated in the subsequent section.

Feature extraction based on GAN

The objective of the third step in the proposed approach is to effectively extract text features related to the content produced through co-occurrence matrices. The proposed method leverages the significant discriminative capability of GAN models. The GAN model employed in the proposed method is based on the UNET33 architecture. Consistent with the general structure of GANs, this neural network comprises both the "generator" and "discriminator" components (Fig. 2).

According to Fig. 2, the generator component of the network is initially fed by a real sample containing noise. The generator then produces a noise-free pattern based on the input sample. Subsequently, the reconstructed pattern and the real sample are provided to the discriminator, which is responsible for evaluating the reconstruction error. The error value is utilized for fine-tuning both the generator and discriminator through the backpropagation process.

In the proposed method, each input sample is described in the form of a square matrix with fixed dimensions. The generator's structure in the GAN model used in the proposed method is depicted in the upper part of Fig. 3. Additionally, the discriminator's structure is shown in the lower part of this figure.

As shown in Fig. 3, the design of the GAN model is based on the encoder-decoder architecture in the general approach of GAN systems. The generator component is responsible for generating artificial patterns, while the decoder section evaluates the reconstruction error.

The GA-SAGAN generator itself is based on the UNet architecture and can be decomposed into encoder and decoder components. The generator's encoder consists of four convolutional layers with a 4×4 dimension. Similarly, the decoder includes four deconvolutional layers of the same size as the encoder, along with two additional output layers of size 0.5. Each of the convolutional layers in this architecture is followed by Batch Normalization (BN) layers and activation functions. The convolutional layers in the generator's encoder use the LeakyReLU activation function, while the ReLU functions are employed for the deconvolutional layers in the decoder. Additionally, the final deconvolutional layer for reconstructing the generator's output uses the hyperbolic tangent activation function.

As mentioned, the purpose of employing the GAN architecture in the proposed method is not to create artificial machine-generated tweets but rather to describe more effective features within this category of messages. For this purpose, the weighted values activated through the last convolution layer in the decoder part of the GAN model are considered as a set of features extracted from the content. Consequently, the model decoder, upon receiving a co-occurrence matrix, yields a weight vector with dimensions of $11 \times 5 \times 128$, which forms the input for the proposed classification model.

It's worth noting that during the GAN model training phase, the Adam optimizer, batch size of 4, and a combination of WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty) and L1 loss functions were used. By employing the WGAN-GP loss function, challenges faced by traditional WGANs, such as gradient descent and mode collapse, are addressed. This is achieved by introducing a gradient penalty term that regularizes the training process, ensuring stable learning. On the other hand, the L1 loss function aims to minimize the absolute difference between the generated output and the ground truth data as closely as possible. This module also provides a higher level of detail to the model in the generated reconstructions. It's important to mention that the proposed method combines the two loss functions as WGANGP + 100L1. Therefore, the L1 loss is multiplied by a coefficient of 100, providing less impact on the WGAN-GP training process. As a result, the WGANGP + 100L1 function used in this model allows for the generation of realistic and accurate outputs while maintaining a stable training process.

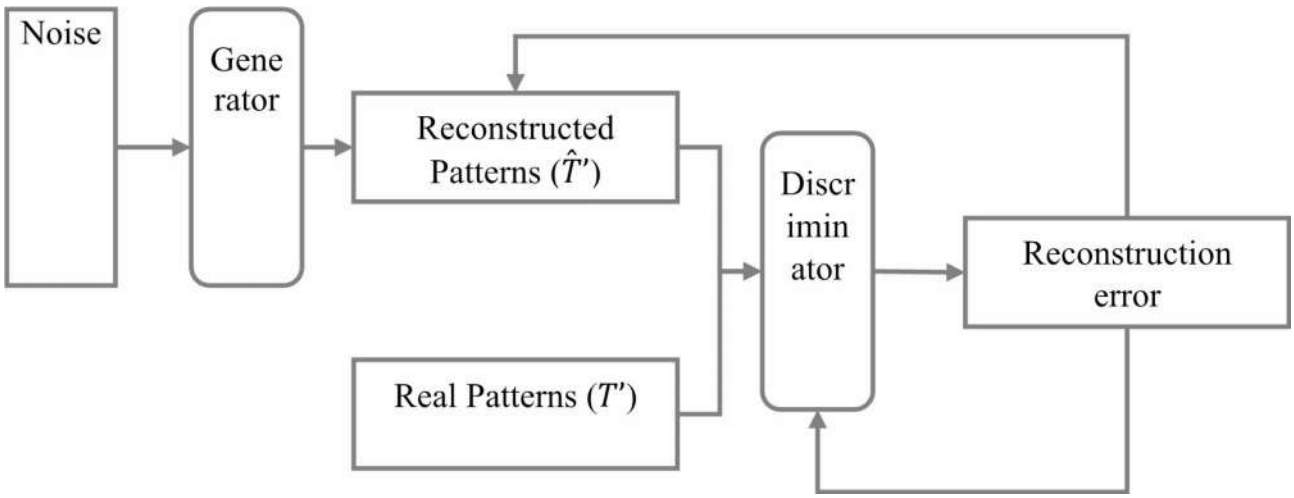

Fig. 2. Architecture of the proposed model for feature extraction.

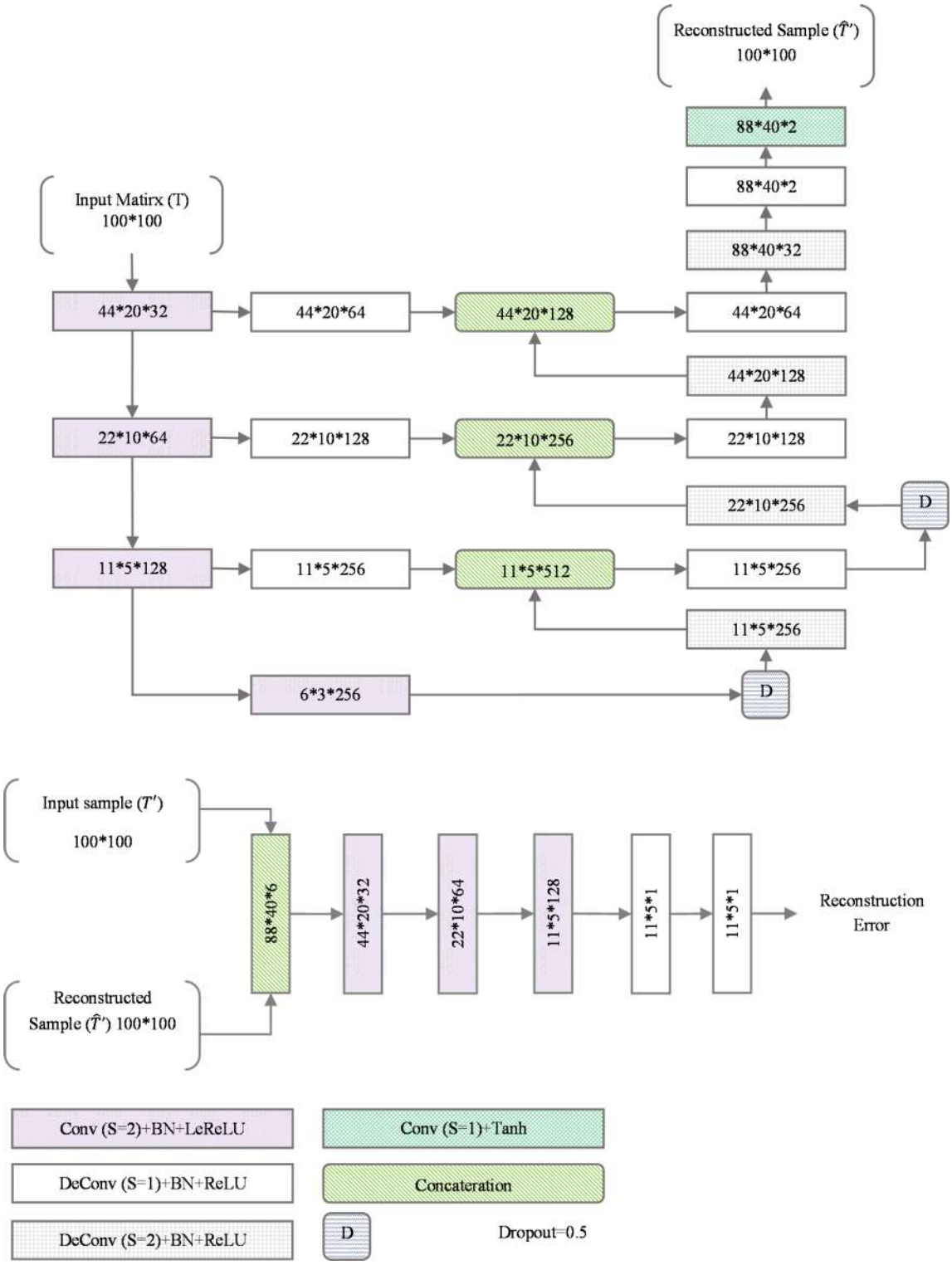

Fig. 3. Generator (top) and discriminator (bottom) structure in the GAN model used in the proposed method.

The GAN's role in feature extraction The GAN model used in our proposed method is central to obtaining features from text co-occurrence matrices for the content generated. The discriminative characteristic of GAN, along with the encoder-decoder structure and the selection of the loss function, allows the model to capture intricate patterns and representations that are useful for classification.

The discriminator: the discriminator part of the GAN is actually a binary classifier that learns the ability to distinguish between the real and generated samples. This process entails learning decision surfaces that defined the boundary between the two classes. The discriminator gets enhanced with time, and in order to pass through

the discriminator the generator comes up with better and better fake data this leads to the extraction of good features from the input data. The discriminator's ability to learn such complex decision boundaries is attributed to the deep structure of the model. The discriminator's layers gradually involve more and more abstract and informative representations of the input data. These learned representations can be also interpreted as a kind of feature engineering where the discriminator learns the best features for discriminating the real and fake samples.

The Encoder-Decoder Architecture: The encoder component of the GAN is responsible for learning raw features from the input data in a hierarchical manner and the decoder component of the GAN is responsible for reconstructing back the input based on such features. Through this process, the model strives to build meaningful representations which could be useful to help the model identify important information present in the text data. We can consider the encoder-decoder structure then in terms of the constraints it as a tool the model applys to the input data, in a compressed format. This condensed representation must contain the most pertinent information required in order to recreate the input data. Therefore, the features retrieved are informative and have a good potential for classification.

The Loss Function's Effect: The learning process of GAN is contingent upon the selection of the loss function. From L1 loss, the signal can be reconstructed with any desired precision and to any level of detail whereas WGAN-GP proceeds with training unimpeded and discourages mode collapse.

In conclusion, it can be seen that the use of GAN in the model performs multiple tasks of feature extraction. They get complex patterns and representations which are very beneficial for classification tasks because of discriminative capability, and the architecture of encoder and decoder and selection of the loss function. The classification model receives much help from the retrieved features and they help the model classify text produced by the artificial intelligence with much accuracy.

RF-based detection

In the final step of the proposed method, an RF ensemble classification model is employed to identify the message author. Consequently, the feature set extracted by the GAN model is used for training an RF model. RF is an ensemble learning model based on decision tree structures, where each decision tree model is trained on a subset of data or a subset of features. The training and construction process for each tree can be independent of other trees in the forest. The RF model used in the proposed method consists of 20 decision tree components. Each decision tree in this ensemble system follows a Classification and Regression Trees (CART), constructed based on the Gini impurity index. Each model is trained based on a random subset of training data samples. The process of constructing CART trees and the formation of the base RF model is detailed in[23], so we refrain from repeating it here. After training the RF model, the majority voting strategy is commonly used to determine the final model output. Considering the performance quality of each decision tree component in the RF and adjusting its impact coefficient on the final model output can enhance the identification accuracy. In the proposed method, each of the 20 CART decision tree components in the RF is assigned a weight value. This weight value represents the impact coefficient of each CART model in the voting process. For instance, if the weight value of a CART model is 3, the output label of this model is counted three times during the voting process for each test sample. Conversely, if this weight value is zero, the CART component's output has no effect on the RF model output. By doing so, more accurate classifiers can receive higher weight values, leading to an effective enhancement of the final detection system's accuracy. It's worth noting that the weight allocation process in the proposed model uses a comprehensive search strategy, allowing each CART model to accept a weight value within the range [0, 7].

Research finding

The proposed approach was implemented using MATLAB 2020a software. In this study, the stratified 10-fold Cross-Validation (CV) was utilized for the proposed and other machine learning models. K-fold CV works by splitting the dataset into k subsets of equal size, and the instances for each subset or fold are randomly selected. The proposed approach has been examined in three operating modes:

- Proposed (GAN + RFW); which refers to the case that combination on GAN and RF with learning components (is described in section "Research methodology") is utilized for classifying instances.
- Proposed (GAN[Discr.]); which refers to the mode that classification of instances is carried out only using the discriminator module of the proposed GAN model. Comparing this mode with the previous case can demonstrate the effectiveness of utilizing a separate classifier for the identification step in the proposed method.
- Proposed (GAN + RF); in this case, the weighing of learning component in the RF model is omitted and classification is carried out using the conventional RFs. Comparing this mode with the Proposed (GAN + RFW) case can reveal the influence of proposed weighting mechanism on the overall performance of the identification system.

Additionally, in order to evaluate the proposed method, a comparison has been made with the SeqXGPT[20], BERT[21], and IDEATE[22] models.

In the following, Table 2 describes the evaluation criteria. The result for the identification process in each sample will be classified to belong to one of the following examples:

- True Positive (TP): Positive (or AI-generated) instances that the model correctly classifies.
- False Negative (FN): Negative (or human-generated) instances that the model misclassifies as positive.
- False Positive (FP): Instances that the model incorrectly labels as positive.
- True Negative (TN): Instances that are correctly detected as negative by the model.

Metric	Equation	Evaluation focus
Accuracy	$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$	Accuracy is the simplest classification metric. It quickly shows how often the model is true by comparing correctly predicted observations to the total data
Precision	$Precision = \frac{TP}{TP+FP}$	Precision is the ratio of correctly predicted positive observations to the total predicted positive observations
Recall	$Recall = \frac{TP}{FN+TP}$	Recall measures the ratio of correctly predicted positive observations to all actual positives
F-Measure	$F - Measure = \frac{2*Precision*recall}{Precision+recall}$	The F-Measure is the harmonic mean of precision and recall

Table 2. Analyzing the evaluation criteria.

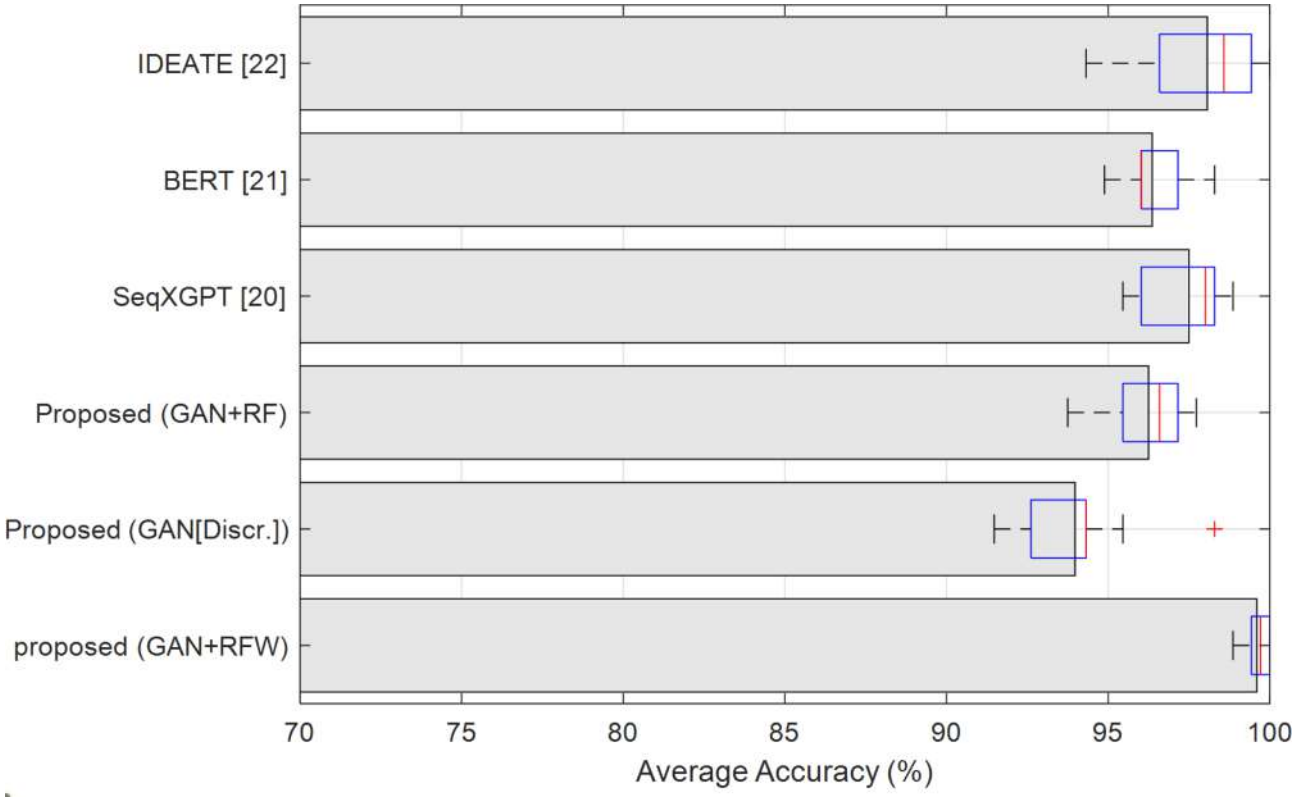

Fig. 4. Performance evaluation of the proposed method and comparative methods in average accuracy.

Figure 4 displays the average accuracy. As can be seen in this figure, the accuracy diagram and the boxplot diagram are both matched to one another. Typically, the box diagram is composed of four parts, with the first to fourth portions displaying the values of accuracy changes in the first to fourth quartiles, respectively. The median accuracy during all folds of CV is represented by the red line that is located in the middle. Both the median and the average accuracy are the same for the method that has been proposed. Additionally, the amount of accuracy changes is in a range that is virtually as narrow as the methods that were examined; in other words, the interval between the first and fourth quartiles is narrower than the intervals that are found in other methods. This indicates that the proposed method was successful in achieving results with a higher level of reliability. This is due to the fact that the results, in addition to being more accurate, are also more closely related to one another than the results obtained from the various iterations of the algorithm.

Figure 5 illustrates the accuracy levels attained during various iterations. The suggested technique clearly surpassed the comparison methods when the accuracy of the various methods is displayed in the 10-fold cross-validation. This indicates that in 90% of the iterations, the suggested technique outperformed the closest-performing method, IDEATE[22]. As a result, we can say that, in general, the suggested method performs better than the comparison methods.

Figure 6 shows the confusion matrix. Rows are true classes; columns indicate the different model predictions. The proposed method (GAN + RFW) has performed better with a 1.5% advantage over comparative method. This figure proves that the proposed method outperforms the compared method in correct classification of both

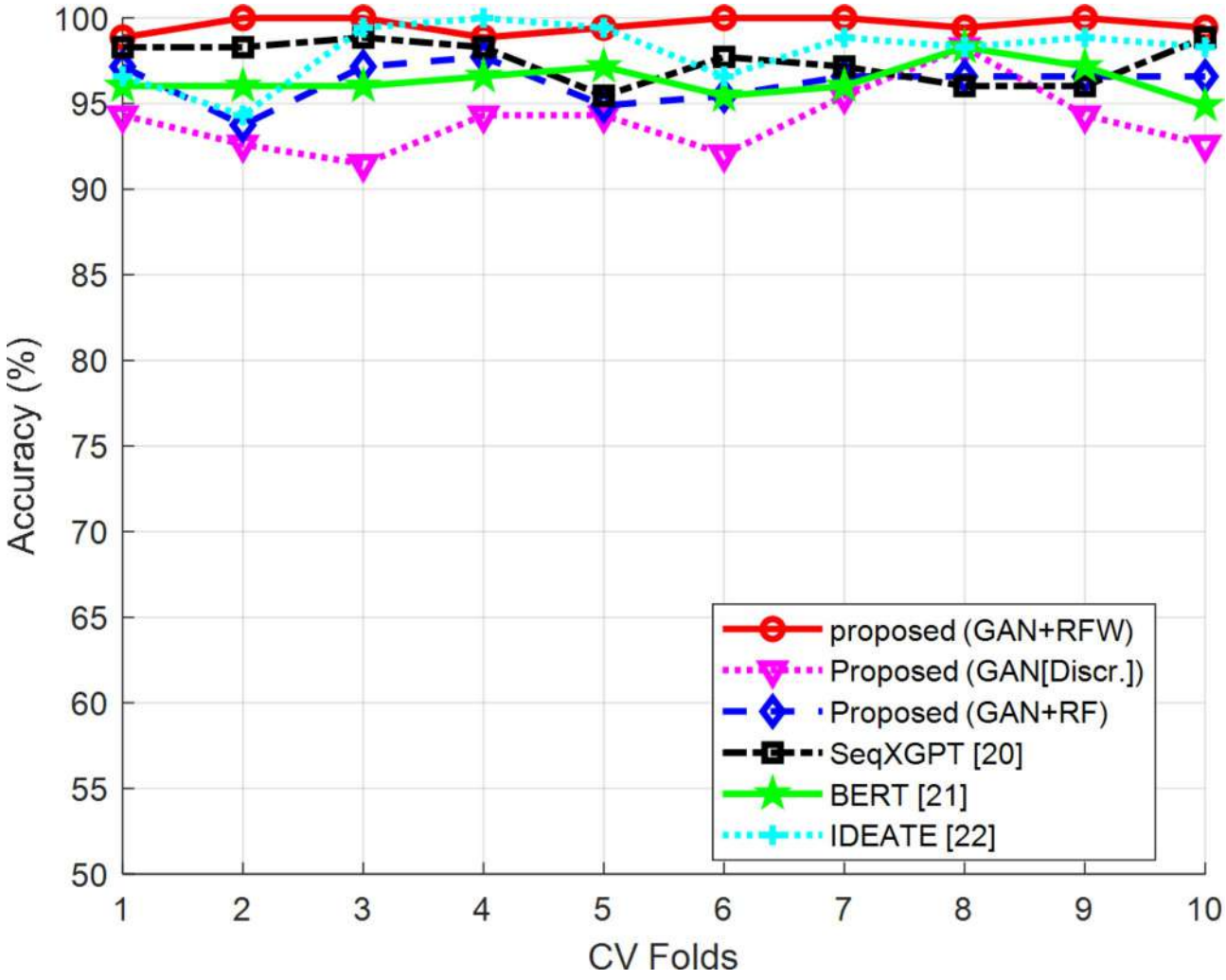

Fig. 5. Evaluation of the proposed method performance and comparative methods in terms of accuracy in 10-folds.

target classes. This means that in addition to providing a more accurate prediction about AI-generated text, our method can also perform better in identifying human-generated texts.

Figure 7 compares the identification methods in terms on precision, recall, and f-measure. In the proposed technique (GAN + RFW), the precision and recall criteria have improved by 0.2% and 3.5% compared to the case (GAN + RF), respectively. This, proves the efficiency of utilizing weighting mechanism in leveraging the performance of RF classifiers. Also, respective superiority of 4.3% and 6.5% in terms of precision and recall compared to the case (GAN[Discr.]); shows the efficiency of utilizing an ensemble-based classifier for the identification task in the proposed method. Furthermore, the proposed method (GAN + RFW) has shown a 1.4% and 1.5% gain in precision and recall criteria compared to the comparison method IDEATE[22]. Additionally, the proposed technique has achieved a remarkable F-Measure value of 0.99%. The results demonstrate that the proposed method has greatly enhanced the classification quality rates for the AI-generated text identification task.

Figure 8 shows the receiver operating characteristic (ROC) curve, showing how the suggested technique reduces false positives and increases genuine positives. The main goal of ROC curve analysis is to find the top-left corner point along the diagonal reference line. The model's discriminative performance is shown by this point's optimal false positive and true positive rate balance. As seen, our strategy performed better than the compared method and this superiority can be attributed to the utilized learning models for feature extraction (GAN) and identification (RFW) steps.

Table 3 summarizes the results obtained through the experiments of this research. Checking this data demonstrates that GAN + RFW is superior in most categories. In precision, proposed (GAN + RFW) performs better with 0.9946. As for recall, this approach got the best score with 0.9978. With 0.9962, the proposed method shows its superior performance using the F-measure criterion. In addition to the highest accuracy of 99.6023%, the proposed approach also reaches the highest AUC (area under the ROC curve) of 0.9978.

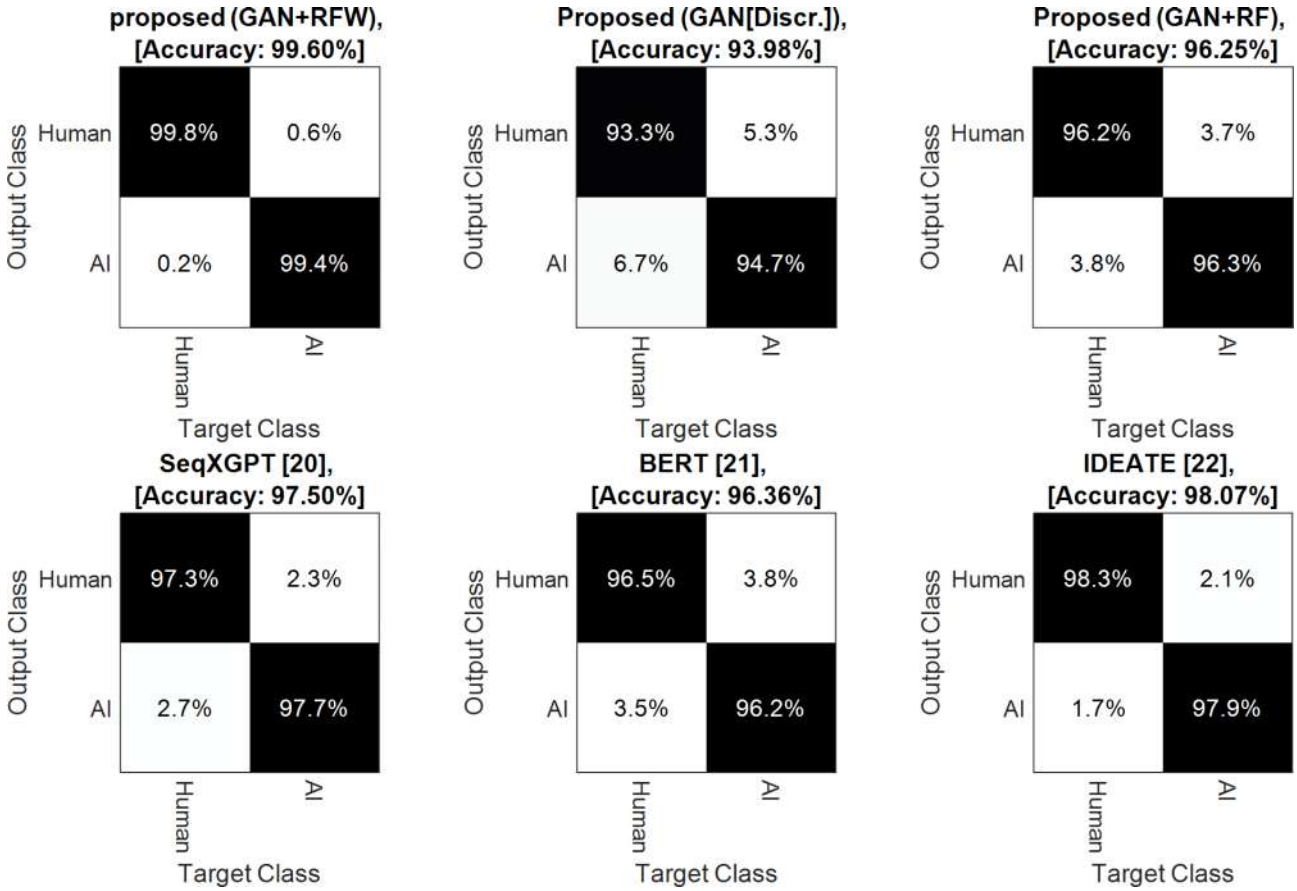

Fig. 6. Performance evaluation of the proposed method and comparative methods in confusion matrix.

Performance analysis of RFW compared to other hybrid classifiers

The purpose of this experiment is to study the effectiveness of the proposed RFW model in text classification compared to existing hybrid tree-based classifiers. For this purpose, in this experiment, the set of features extracted by the proposed GAN model has been classified by different algorithms and the results have been compared based on different criteria. It should be noted that the cases compared in this analysis are only different in terms of classification step, and in other words, all cases are fed through the same features. The first classifier to be compared is Gradient Boosting, which is based on 100 decision tree classifiers. In this model, the learning rate parameter is equal to 0.01 and the maximum depth is equal to 3. The second compared hybrid classifier is XGBoost, whose number of binary tree components is determined based on cross-validation strategy. In the implementation of this model, the parameters of the maximum depth and the minimum weight of the child are set as 5 and 10, respectively. The evaluation criterion of the model is the validation error rate and the learning rate is 0.01. Table 4 shows the results of these tests.

As it is clear from the results presented in Table 4, the proposed RFW model leads to achieving higher accuracy in distinguishing the AI-generated texts from the texts written by humans. In addition to the accuracy criterion, this superiority was also evident in other compared criteria, which confirms the optimal performance of this model. As shown before, the RFW model increases the accuracy by 3.35% and f-measure by 3.2% compared to the basic random forest model. This superiority compared to the Gradient Boosting model is equal to 1.5 and 1.4%, respectively. In contrast, the performance of the proposed RFW with the XGBoost model does not show significant differences. Both of these models use the weighting strategy in different ways in order to improve efficiency. The favorable performance of RFW can be attributed to the allocation of weights to tree components. Although this strategy results in a significant increase in computational load; But based on the obtained results, it leads to improving the performance of the classification model.

Analyzing the influence of Tweet length on the performance of the model

To assess the effect of the maximum number of characters in the tweet on the performance of our model, we carried out tests on the different maximum tweet lengths of 50 to 250 characters. We evaluated the performance of the proposed model of GAN for feature extraction and RFW for classification on each of the tweets' length. The results of this experiment are summarized in the Fig. 9.

From the Fig. 9, it is clear that the accuracy increases as the maximum tweet length increases. Also, the same trends are observable for other operating modes of the proposed method. This suggests that maybe there is extra information contained in the longer tweets that can be used in the distinction between the texts produced

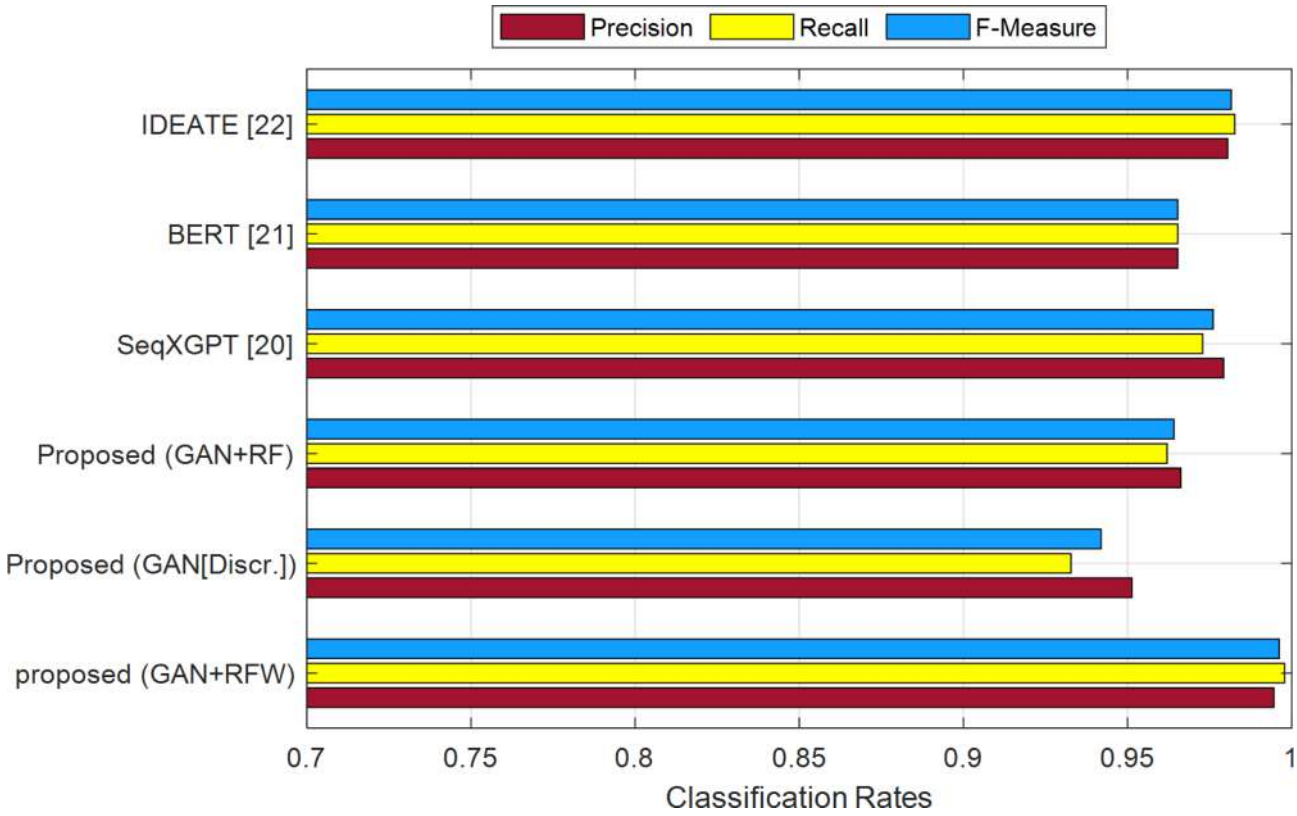

Fig. 7. Performance evaluation of the proposed method and comparative methods in terms of precision, recall, and f-measure.

by human and those produced by AI. The observed trend of increasing accuracy with longer tweets can be attributed to several factors:

- Information Density: Longer tweets frequently have more information, which could give the model more hints on how to tell AI-generated material from human-generated content. Longer tweets, for instance, could have more context, precise information, or subtle phrases that show human authorship.
- Feature Richness: Longer tweets also encompass more features such as the vocabularies, grammars and style that can be exploited by the model. Some of these features can be helpful for classification when combined with features acquired from the GAN.
- Reduced Noise: Shorter tweets may contain a lot of noise or random variation that makes a certain tweet difficult to classify as is seen in Fig. 9. Tweets with more information may be less affected by noise because of the length of the message.

But what needs to be mentioned is, that there is an interaction effect revealed whereby the performance of the presented models is not a simple function of the length of the Compilation of Tweets. There could be a point of inflection for their case, meaning that the value of adding extra characters to the tweet may not add value if the extra characters are redundant or meaningless. However, it should be pointed out that, due to the smaller number of instances for shorter tweet lengths, the generalization of the results of this analysis may only be partial. A larger dataset would be a better way of making a more accurate comparison between the length of the tweets and the accuracy of the information relayed.

Discussion, implications, and limitations

This section discusses the validity and reliability of the proposed method, the implication of the study, and the weakness of the method. We report on the results of our experiments and consider further research opportunities.

Performance analysis

The method presented in this paper, GAN feature extraction followed by weighted random forest classification, outperforms some of recent works such as SeqXGPT[20], BERT[21], and IDEATE[22] and improves the accuracy by at least 1.5%. In Table 3, the performance comparison of all the approaches that have been implemented demonstrate that the proposed method has the highest accuracy of 99.60%, precision of 0.9946, recall of 0.9978, F-measure of 0.9962, and AUC of 0.9978. This approach could improve the performance of the model from various aspects such as precision, recall, and f-measure. The improved performance of our method can be attributed to several factors:

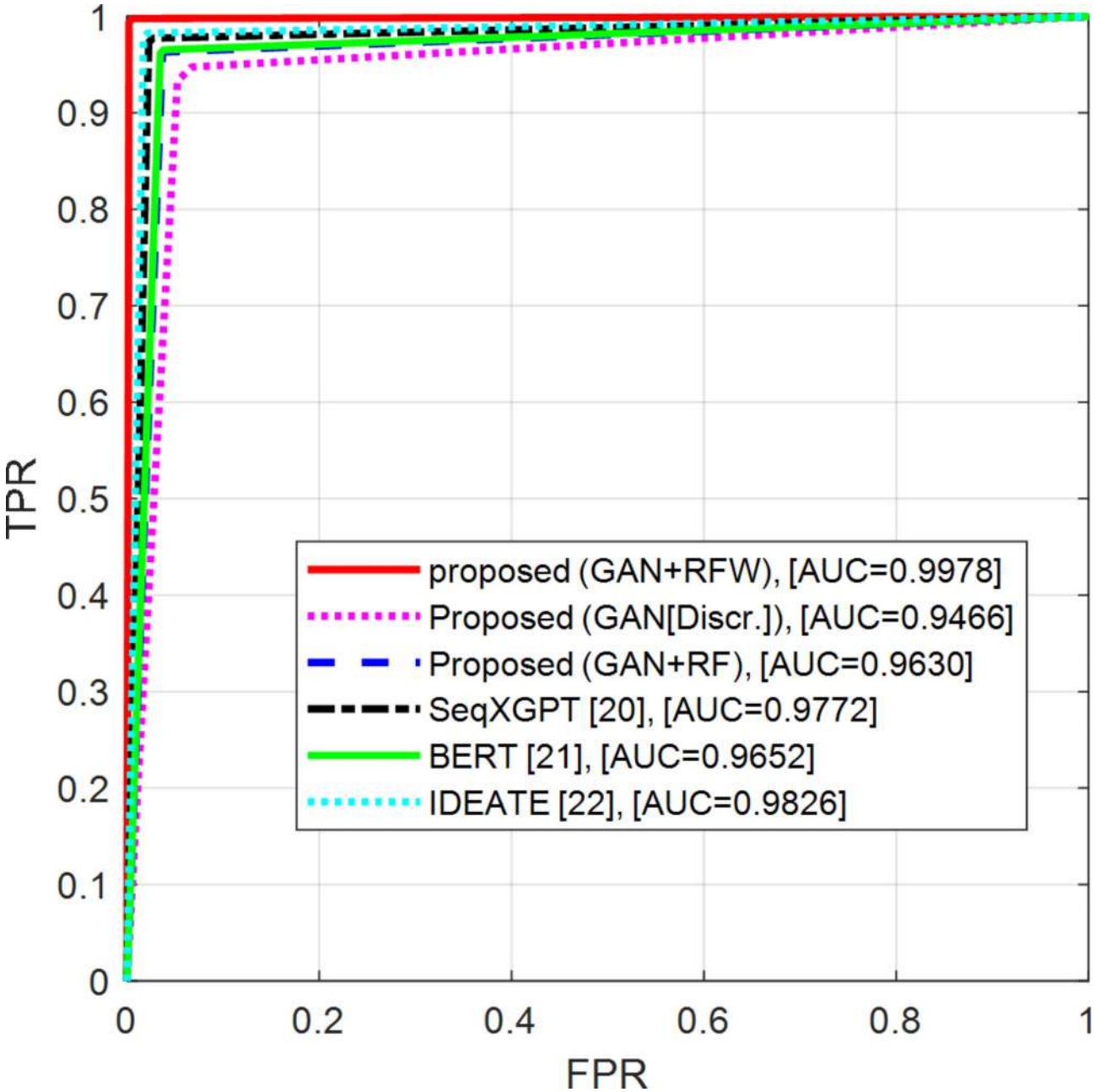

Fig. 8. Performance evaluation of the proposed method and comparative methods in ROC curve.

Models	Precision	Recall	F-measure	Accuracy	AUC
Proposed (GAN + RFW)	0.9946	0.9978	0.9962	99.6023	0.9978
Proposed (GAN[Discr.])	0.9513	0.9328	0.9419	93.9773	0.9466
Proposed (GAN + RF)	0.9662	0.9620	0.9641	96.2500	0.9630
SeqXGPT[20]	0.9793	0.9729	0.9761	97.5000	0.9772
BERT[21]	0.9653	0.9653	0.9653	96.3636	0.9652
IDEATE[22]	0.9805	0.9826	0.9816	98.0682	0.9826

Table 3. Comparison of evaluation criteria of the proposed method with comparative approaches.

Models	Precision	Recall	F-measure	Accuracy	AUC
RFW	0.9946	0.9978	0.9962	99.6023	0.9978
XGBoost	0.9935	0.9967	0.9951	99.4886	0.9967
Gradient boosting	0.9826	0.9805	0.9815	98.0682	0.9809
Conventional RF	0.9662	0.9620	0.9641	96.2500	0.9630

Table 4. Comparison of the proposed RFW with the existing hybrid tree-based classifiers.

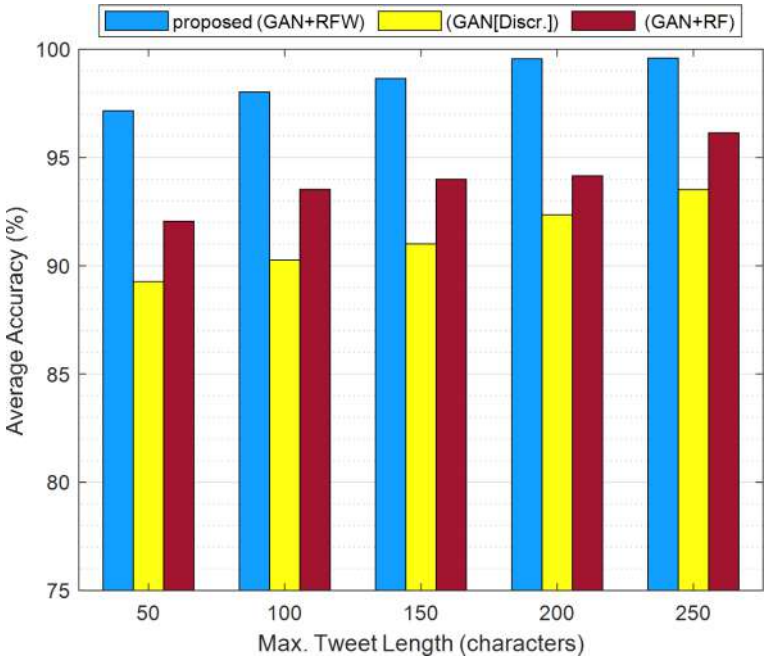

Fig. 9. Influence of Tweet length on the accuracy of the model.

- Effective Feature Extraction: The GAN successfully filters the relevant features inherent in the textual data and captures the differences between the text written by people and AI.
- Weighted Random Forest: The RFW classifier integrates the features well to yield right predictions. The algorithm used in RFW known as weighting mechanism assists in enhancing the performance since it allocates suitable weights to the individual trees. As shown in section "Performance analysis of RFW compared to other hybrid classifiers" and Table 4, the proposed weighting mechanism could improve the classification accuracy by 3.35%.
- Synergistic Integration: GAN and RFW when used together produce a strong and efficient model for detecting text written by an AI. This combination, leverages the accuracy by 5.62% compared to the case which the discriminator of GAN is used for detection.

Practical implications
Our research has several practical implications:

- Improved AI-Generated Text Detection: The proposed method can be used for enhancing the accuracy of AI text detection that is necessary for various applications, such as combating faux news and spam.
- Enhanced Content Moderation: The current approach of ours can serve as an integration to content moderation services which aid in finding content created by AI and is against the rules or terms of a community.
- Improved Language Model Evaluation: The described approach can be applied to evaluate the quality and the closeness to real text generated by developing AI algorithms which will be useful for language model developers.

Limitations and future directions
The proposed method has several limitations, based on which future directions can be determined:

- Dataset Dependence: The performance of the proposed model depends on the characteristics of the dataset used for its training and evaluation.
- Evolving AI-Generated Text: As large language models progress, it becomes more difficult to identify the texts produced by them. This increase in complexity will also affect the performance of the proposed model.

- Diversity of Data: Although, the idea that laid the foundation for the proposed model's efficiency could be limited to the platforms or the type of text in the dataset. Thus, its effectiveness could be rather doubtful in case it is applied to other platforms or types of writing.
- Computational Cost: Training GANs besides using multiple learners in RFs is a complex computational process and need a considerable time. This could somewhat reduce the feasibility of using the approach in large scale applications.

Future research could explore the following directions:

- Larger and More Diverse Datasets: The robustness and generalizability of the proposed method can be examined on the larger and varied datasets.
- Optimized GAN Architectures: Future research could focus on replacing the GAN model with more efficient and recent architecture that can help reduce the computational expenses.
- Adaptability to New Language Models: This direction includes exploring the way of modifying the proposed approach to detect AI generated text from newer and stronger language models.
- Real-Time Detection: Developing real-time detection capabilities in the proposed approach to enable timely identification of AI-generated text in dynamic environments.

Conclusion

In the paper, it is proposed to apply sophisticated generative AI models so as to raise the degree of accuracy of machine learning algorithms in the identification of AI-generated text. The technique uses the functionality for generation to further incorporate ensemble learning strategies with a view to perfectly characterize the generated content. There are four steps in the process: first, processing initial text for the purpose of eliminating extraneous data; feature engineering and encoding that will transform texts into a square matrix defining word associations and significance; feature extraction via a GAN for the classification of texts by origin; and detection via RF algorithms with features derived from the GAN discriminator. It makes a very impressive differentiation between human and AI-authored texts, with an average accuracy of 99.60%, meaning it improves on previous techniques by 1.5%.

Despite the acceptable performance of the proposed model in this research, this approach faces limitations:

- The efficiency of the proposed model might be constrained to the platforms or the kind of text contained in the dataset. Therefore, its performance might be questionable in case of applied to other platforms or writing styles.
- Training GANs along with the use of multiple learners in RFs is computationally expensive and time-consuming. This could somewhat reduce the feasibility of using the approach in large scale applications.
- The proposed model is text based (tweets) and might not be easily translatable to other data types such as images or videos generated by AI. These might need different feature extraction methods and may even need new architectures for the models.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 19 July 2024; Accepted: 1 November 2024
Published online: 26 November 2024

References

1. Ricker, J., Assenmacher, D., Holz, T., Fischer, A. & Quiring, E. AI-generated faces in the real world: a large-scale case study of twitter profile images. arXiv preprint arXiv:2404.14244 (2024).
2. Meng, Y., Fang, G., Yang, J., Guo, Y. & Wang, C. C. *Spring-IMU fusion-based Proprioception for Feedback Control of soft Manipulators* (IEEE/ASME Transactions on Mechatronics, 2023).
3. Mu, P., Zhang, W. & Mo, Y. Research on spatio-temporal patterns of traffic operation index hotspots based on big data mining technology. In *Basic and Clinical Pharmacology and Toxicology* (Vol. 128, 185–185) (Wiley, 2021).
4. Imani, A. et al. Glot500: Scaling multilingual corpora and language models to 500 languages. arXiv preprint arXiv:2305.12182. (2023).
5. Bernacki, B. E., Johnson, T. J. & Myers, T. L. Modeling thin layers of analytes on substrates for spectral analysis: use of solid/liquid n and k values to model reflectance spectra. *Opt. Eng.* **59**(9), 092005 (2020).
6. Szabó, F. J. Comparison of Programming Ansys and COSMOS/M finite element systems. *Design of Machines and Structures: A Publication of the University of Miskolc* **12**(2), 110–119 (2022).
7. Jiang, Y., Bugby, S. L. & Lees, J. E. PMST: A custom Python-based Monte Carlo Simulation Tool for research and system development in portable pinhole gamma cameras. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* **1061**, 169161 (2024).
8. Tran, Q. D., Nguyen, V. Q., Pham, Q. H., Nguyen, K. B. & Do, T. H. Vietnamese AI Generated Text Detection. arXiv preprint arXiv:2405.03206 (2024).
9. Wang, R., Li, Q. & Xie, S. DetectGPT-SC: improving detection of text generated by large language models through self-consistency with masked predictions. arXiv preprint arXiv:2310.14479 (2023).
10. Bhattacharjee, A. & Liu, H. Fighting fire with fire: can ChatGPT detect AI-generated text? *ACM SIGKDD Explor. Newsl.* **25**(2), 14–21 (2024).
11. Ghosal, S. S. et al. *A Survey on the Possibilities & Impossibilities of AI-generated Text Detection* (Transactions on Machine Learning Research, 2023).
12. Aguilar-Canto, F., Cardoso-Moreno, M. A., Jiménez, D. & Calvo, H. GPT-2 versus GPT-3 and Bloom: LLMs for LLMs Generative Text Detection. In IberLEF@ SEPLN (2023).
13. Lokna, J., Balunovic, M. & Vechev, M. Human-in-the-loop detection of AI-generated text via grammatical patterns.

14. Ghosal, S. S. et al. Towards possibilities and impossibilities of ai-generated text detection: a survey. arXiv preprint arXiv:2310.15264 (2023).
15. Abburi, H. et al. A simple yet efficient ensemble approach for Ai-generated text detection. arXiv Preprint arXiv:231103084 (2023).
16. Suvra Ghosal, S. et al. Towards possibilities and impossibilities of AI-generated text detection: a survey. arXiv–2310 (arXiv e-prints, 2023).
17. Akram, A. An empirical study of Ai generated text detection tools. arXiv Preprint arXiv:231001423 (2023).
18. An, B. AI-generated text detection: challenges and future directions. *Int. J. Asian Lang. Process.* **33**(02), 2330002 (2023).
19. Zhang, Y. et al. Detection vs. anti-detection: is text generated by AI detectable? In *International Conference on Information* 209–222 (Springer Nature Switzerland, 2024).
20. Wang, P. et al. SeqXGPT: sentence-level AI-generated text detection. arXiv Preprint arXiv:231008903 (2023).
21. Wang, H., Li, J. & Li, Z. AI-generated text detection and classification based on BERT deep learning algorithm. arXiv preprint arXiv:2405.16422 (2024).
22. Wang, Q., Zhang, L., Guo, Z. & Mao, Z. IDEATE: detecting AI-generated text using internal and external factual structures. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* 8556–8568 (2024).
23. Daniya, T., Geetha, M. & Kumar, K. S. Classification and regression trees with Gini index. *Adv. Math. Sci. J.* **9** (10), 8237–8247 (2020).

Author contributions
Yang Hui wrote the main manuscript text. Yang Hui reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to Y.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.