International Journal of Scientific Research and Management (IJSRM)
||Volume||14||Issue||09||Pages||3053-3063||2026||
Website: https://ijsrm.net ISSN (e): 2321-3418

Combining probabilistic features and semantic features for AI-Generated text detection

Yang Yu[1], Wang Gao[1]
[1]*School of Artificial Intelligence, Jianghan University, Wuhan, China*

Abstract: The proliferation of Large Language Models (LLMs) such as ChatGPT and Gemini has resulted in a surge of AI-generated text across various domains. However, the widespread use of this technology raises concerns regarding the generation of misinformation and malicious content. To address this challenge, we propose a novel AI-generated Text Detection model combining Probabilistic and Semantic features (ATDPS). Our model extracts semantic features using a pre-trained language model and combines them with probabilistic features generated by multiple LLMs. A temporal convolutional network is employed to process sequence probabilistic features, effectively capturing temporal characteristics within the text. To ensure data coherence and diversity, our dataset includes text generated by a variety of LLMs, including the latest models like GPT-4. Experimental results demonstrate ATDPS's superior performance over existing baselines in terms of accuracy, precision, recall and F1 score, highlighting its potential and effectiveness in detecting AI-generated text.

Keywords: AI-generated text detection, large language models, pre-trained language models, temporal convolutional networks, semantic features, probabilistic features

1 Introduction

The rapid advancement of Large Language Models (LLMs) such as PaLM, ChatGPT, and Gemini has significantly contributed to the proliferation of AI-Generated Text (AIGT) in various domains, as highlighted by recent studies [1], [2], [3], [4]. These models demonstrate remarkable capabilities, illustrating their transformative potential across a multitude of applications, such as information extraction and sentiment analysis [5], [6], [7]. However, the rise of AIGT has also sparked significant concerns regarding its potential misuse [8]. The ability of these models to generate text that is both realistic and human-like poses serious challenges, particularly in relation to misinformation, disinformation, and the generation of malicious content [9], [10]. As illustrated in Table 1, the task of distinguishing between text produced by various LLMs and that written by humans is often not straightforward, even for trained experts.

Table 1 Illustrative examples of AI-generated and human-written text samples.

Model	Text
GPT-2	The addition of sodium chloride (1 mmol/l) did not influence their growth and (b) LPS treated with salt (40 mm) significantly ($p<0.01$) induced an IC50 value of 11 nmol TNFα/mL.
GPT-J	The production of nitric oxide (NO), prostaglandin-E (PGE2) and prostaglandin-D-synthetase (PG-D synthase) were measured by nitrate reductase assay.
GPT-Neo	The results from ELISA demonstrated that the highest levels of IL-8 and ICAM-1 and the lowest level of CXCL10 production by LPS.
LLaMA	The addition of extra NaCl resulted in significant increase of LPS-induced nitric oxide production in a concentration-dependent manner as well as of LPS-induced INF-β production in a dose-dependent way.
Human	NaCl had no influence on the apoptosis and proliferation of ARPE-19. Addition of 40 mm NaCl significantly induced IL-6 and MCP-1 production but had no effect on IL-8 secretion.

The 2024 U.S. presidential election serves as a pertinent example of the potential dangers associated with AIGT[1]. Reports indicate that political campaigns are increasingly employing AI-generated content for fundraising emails and promotional materials. Notably, the Republican National Committee has purportedly released a video featuring AI-generated imagery that depicts a dystopian scenario following President Biden's re-election. Such manipulative content not only risks exacerbating societal biases but also

threatens the integrity of fair competition and deepens voter entrenchment in polarized information silos. These instances underscore the urgent need for developing robust methodologies for the detection and regulation of AIGT, aimed at mitigating its potential role in spreading misinformation and facilitating cybercrime.

Current AIGT detection methods can be broadly classified into black-box and white-box approaches [11]. Black-box detection involves training classification models, such as GPTZero [12], using a dataset comprising samples from both human and machine-generated texts. This method identifies generated content by analyzing various metrics, including perplexity and burstiness [13], [14]. In contrast, white-box detection methods have comprehensive access to the LLMs, enabling them to track and detect generated text by manipulating the model's output behavior or embedding watermarks within the text. Typically developed by LLM creators, these detectors tend to provide more accurate identification of AIGT.

One prominent white-box detection technique leverages the probabilistic features of language models. For instance, Wang et al. utilize white-box models to compute the log probability of each word within a text sequence [15]. These log probabilities are subsequently transformed into temporal features resembling waveforms, which are processed using Convolutional Neural Networks (CNN) and self-attention mechanisms. The models are then trained via a sequence labeling approach, assigning the most frequently occurring word category to the corresponding sentence.

Probability-based methods capitalize on a language model's inherent capability to represent text generation [16]. By analyzing the probability distributions of words within a text sequence, these methods can identify discrepancies between AI-generated and human-written texts. Specifically, perplexity serves as a key metric in this analysis and is mathematically defined as the negative average of the sum of the log probabilities of all words in the text. An inverse relationship exists between perplexity and log probability; a model that assigns high probability estimates to a given text will yield low perplexity, indicating strong predictive capability.

However, as LLMs continue to evolve, the perplexity of AIGT increasingly mirrors that of human-written texts, complicating reliance on probability features alone for effective text identification. Moreover, the performance of probability-based methods can vary significantly across different text types and styles, as specialized domains often exhibit distinct linguistic patterns that influence the accuracy of probability assessments [17]. Consequently, these methods frequently struggle to achieve high accuracy when used in isolation, necessitating their integration with additional metrics, such as grammar error rates and contextual consistency, to enhance detection effectiveness. These limitations suggest that while probability-based approaches show promise, they are insufficient to fully address the complexities and diversities inherent in contemporary AI-generated content.

To address these challenges, we propose a novel model called the AI-generated Text Detection model combining Probabilistic and Semantic features (ATDPS). This model integrates semantic features into the detection framework and utilizes a pre-trained language model to extract these features. By analyzing the semantic content of the text, our model gains a deeper understanding of the underlying meanings, thereby enhancing its ability to differentiate AIGT from human-written text. Unlike traditional methods, which primarily rely on surface features such as word frequency and sentence structure, our approach focuses on capturing the subtle nuances and deeper meanings inherent in the text. This capability is crucial for identifying AIGT that closely mimics human language. By leveraging the pre-trained language model for semantic feature extraction, our model is better equipped to detect misleading and deceptive content, ultimately curbing its dissemination on social media and news platforms. This proactive measure contributes to mitigating the negative impacts on public perception and decision-making.

Furthermore, we employ Temporal Convolutional Networks (TCN) [18], which excel in processing sequential data and effectively capturing temporal features within text. Traditional CNNs often overlook the temporal dynamics present in textual data. In contrast, TCNs, with their specialized architecture, adeptly model temporal dependencies and contextual relationships. This improvement allows our model to maintain efficient detection capabilities even when facing complex texts, such as long articles, dialogues, and technical documents, effectively curbing the spread of misleading AI-generated content in these fields and protecting the authenticity and reliability of information.

Finally, we expand the size and diversity of the training dataset from SnifferBench [19]. The original dataset comprised texts generated by models such as GPT-3 and LLaMA-2. Building upon this foundation, we incorporated texts produced by GPT-4, thereby enhancing the model's generalization capabilities. This expansion enables the ATDPS model to achieve high stability and accuracy when addressing the latest forms of AIGT. As AI technology continues to evolve rapidly, new generations of language models are emerging, making their outputs increasingly indistinguishable from human-written content. By integrating texts generated by GPT-4, our model is positioned to swiftly adapt to and identify the latest

AI-generated content, preventing these new variants of misinformation from causing significant harm to public opinion, market decisions, and personal privacy.

Our contributions in this paper are as follows:

We propose the ATDPS model, which integrates both probabilistic and semantic features. By analyzing the semantic content, our model achieves a deeper comprehension of text meanings, significantly improving its ability to detect AIGT. The incorporation of semantic features enables ATDPS to capture subtle text differences more accurately than traditional methods, improving performance particularly in cases where AIGT closely mimics human writing.

We introduce TCNs into the ATDPS model for encoding temporal probabilistic features. TCNs are particularly effective in handling sequential data, enabling the model to capture temporal dependencies and contextual relationships within the text. To our knowledge, we are the first to employ TCNs for processing temporal features while simultaneously integrating probabilistic and semantic features for AIGT detection.

The experimental results indicate that our proposed ATDPS model outperforms existing state-of-the-art models in terms of accuracy, precision, recall and F1 score. This advancement underscores the effectiveness of our approach and its potential applicability in real-world scenarios.

2 Related work

The proliferation of LLMs has dramatically enhanced their performance across various language-related benchmarks, culminating in their ability to generate highly convincing textual content [20], [21], [22], [2]. A pioneer in this field, the GROVER model, was specifically designed to generate realistic news articles [23]. Human evaluators determined that GROVER's output was indistinguishable from human-written content in terms of trustworthiness when generating propaganda text, even surpassing human-created content in certain aspects. The trajectory of NLP suggests that these LLMs will continue to evolve, posing significant challenges in terms of authenticity and regulation [24], [25]. The prevalence of AI-rewritten news articles has highlighted fundamental issues, prompting a surge in research dedicated to detecting AIGT [26].

Existing detection efforts primarily classify AIGT as a binary classification problem, employing neural network-based detectors [14], [27], [28], [29], [30]. OpenAI's fine-tuning of a RoBERTa-based model [31] for GPT-2 detection exemplifies this approach. However, this method necessitates fine-tuning for each new LLM, hindering its scalability. Zero-shot AIGT detection, as introduced by DetectGPT, offers a potential solution [14], [32], [13]. Nevertheless, the reliance on neural networks makes these methods susceptible to adversarial and poisoning attacks [33], [34]. To address these limitations, watermarking techniques have been proposed, embedding specific patterns into AI-generated text for easier detection [35], [36]. While these watermarks are generally imperceptible to humans, they can be exploited for detection purposes. Krishna et al. proposed an information retrieval-based detector, which stores LLM outputs in a database, raising serious privacy concerns [37].

Compared to existing work such as SeqXGPT, which leverages CNNs over log-probabilities, our main innovation lies in the integration of semantic features with probabilistic signals. By combining token-level probability vectors with contextual semantic embeddings, and modeling their interactions using a Transformer, our approach enables a more comprehensive representation of both lexical likelihood and contextual coherence. This fusion strategy significantly improves detection of AI-generated text, particularly in challenging cases where either signal alone is insufficient.

3 Methodology

In this section, we first define the task of detecting AIGT. Next, we introduce the structure of the ATDPS model. Finally, we provide a detailed description of the model's main modules.

3.1 Problem Definition

Given a piece of text (such as international news articles, tweets, etc.) along with its corresponding label (indicating whether the text was generated by a language model or written by a human), the AIGT detection task aims to predict the source of the text. Formally, we define the problem as follows:

Input: A sequence of text sampled from various domains, which may be either human-written or generated by a language model.

Output: A binary classification indicating whether the input text is human-written or AI-generated.

3.2 Model Architecture

The architecture of the ATDPS model is illustrated in Figure 1. Within this framework, for a given text α containing β words, we first extract probability features $S=[w1,w2,...,wn]$ from the basic features using TCNs. Next, we employ the pre-trained language model BERT [38] to extract semantic features n from the text $Fp=[x1,x2,...,xn]$. Subsequently, the extracted probability features Fs and semantic features S are concatenated to form a unified feature set. These concatenated features are then processed through a Transformer to capture additional contextual information. Finally, the output is passed through a linear classification layer to determine the source of the text, indicating whether it was generated by a language model or written by a human.

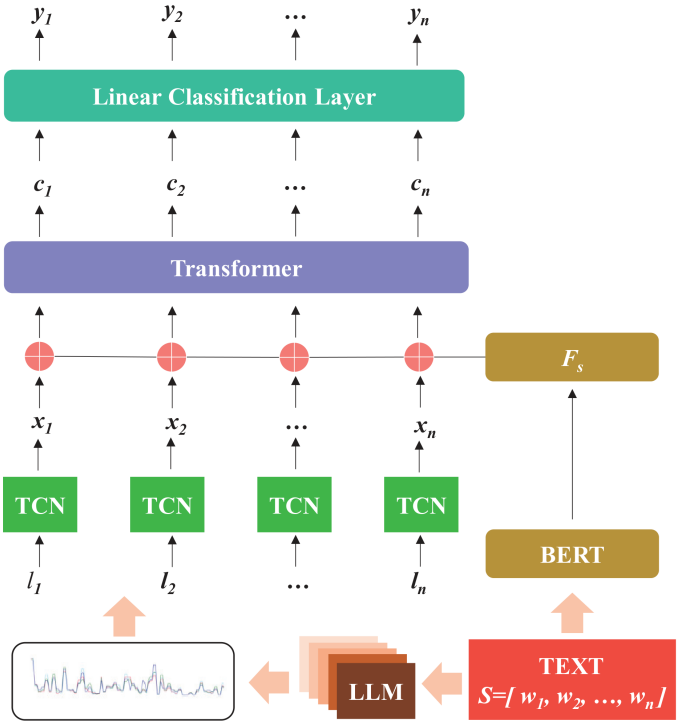

Figure 1 *The architecture of the ATDPS model.*

The ATDPS model comprises three key stages: (1) feature extraction and alignment; (2) feature encoding and fusion; and (3) linear classification. The following sections will detail each of these stages.

3.3 Feature Extraction and Alignment

Mitchell et al. highlight that contrastive features derived from token-wise probabilities, as well as the average per-token log probability, are beneficial for identifying AIGT [14]. Building upon this insight, the ATDPS model selects a list of token-level log probabilities extracted from a series of language models as the basic features for the input text.

Specifically, let Fp represent the input text containing Fs tokens. Given a series of language models S=[t1,t2,...,tm], the log probability m of the θ=[θ1,θ2,...,θM]-th token lθi(ti) calculated based on the i-th language model ti can be expressed by the following equation:

$$l_{\theta_i}(t_i) = logp_{\theta_i}(t_i|t_{(i)}),$$

where i denotes the probability function under the language model θi, and pθi(·) represents the sequence of tokens preceding θi. By applying these ti) language models, we can generate a list of token-level log probabilities for each token in the input text ti. This results in M log probability values for each token, corresponding to the respective language models.

To mitigate the challenges posed by varying tokenization methods and granularities across different LLMs, we devise a token-to-word alignment strategy. This approach addresses potential discrepancies arising from the tokenization process, ensuring a more consistent representation of the input text. For the input text S comprising M words, ATDPS aligns token-level probabilistic features to their corresponding general words. In instances where a word S is aligned to multiple tokens within n, ATDPS averages the log probabilities of these tokens to provide a comprehensive representation. Conversely, if a token aligns to multiple words, the same value is assigned to each of these words. This alignment process enables the extraction of word-level log probabilities w from the original token-level log probabilities S. By concatenating these word-level log probability lists, we construct the basic feature set lθi(w). The basic feature vector lθi(t) of the L=[l1,l2,...,ln]-th word li is represented as:

$$l_i = [l_{\theta_1}(w_i), l_{\theta_2}(w_i), \ldots, l_{\theta_M}(w_i)].$$

To obtain these basic features, we employ a suite of white-box language models, including GPT-2, GPT-3, GPT-Neo, GPT-J, and LLaMA-2. These models provide diverse perspectives and enhance the robustness of the probability feature extraction process.

To address the limitations of probability-based features in capturing the nuances of contextual meaning, we incorporate semantic features derived from the BERT model. By leveraging left and right contextual information, BERT is adept at understanding the meaning of words and their relationships within a given text. For the input text i, the BERT encoder computes attention weights for each token wi using the following equation:

$$Attention(t_i) = softmax\left(\frac{Q_i K^T}{\sqrt{d_k}}\right)V,$$

where S represents the query vector for the ti-th token Qi, i is the key matrix for all tokens, ti is the value matrix for all tokens, and K is the dimension of the key used for scaling. The BERT encoder subsequently generates a contextual representation for each word in the input text V. Typically, the hidden state of the final layer is employed as the semantic feature dk:

$$F_s = BERT(S)$$

In our implementation, the BERT embeddings used for extracting semantic features are fine-tuned during the supervised training process. This allows the model to adapt the semantic representations to the AIGT detection task, rather than relying solely on static, pre-trained representations.

3.4 Feature Encoding and Fusion

The unique nature of our basic features, consisting of lists of word-level probability features, precludes the direct application of existing pre-trained models. To effectively address AIGT detection, it is necessary to design a novel model architecture

that is capable of performing sequence labeling based on these specialized features. The word-level probability lists extracted from different LLMs for the same sentence may exhibit variations due to disparities in parameter scales, training data, and other factors. We propose interpreting these lists as features that reflect a model's understanding of semantic and syntactic structures. More complex models, capable of learning intricate language patterns and syntactic structures, are likely to generate higher probability scores. Additionally, the actual probability lists may be subject to uncertainty arising from sampling randomness and other factors.

Considering these properties, we draw an analogy between the temporal nature of probability lists and the wave-like nature of speech signals. Inspired by the successes of TCNs in speech processing, we have chosen to employ a TCN architecture for encoding the sequential patterns in basic features. TCNs are well-suited for handling sequential data with temporal dependencies, making them a natural choice for processing the word-level probability lists derived from the input text. Let S represent the convolutional kernel for a given layer of the TCN, and the convolution operation for the Fs-th word F=[f1,f2,...,fk] can be expressed as:

$$F(w_i) = \Sigma_{j=1}^{k} f_j l_{\theta_{i-(k-j)d}(w_i)},$$

where i represents the convolved output for word wi, F(wi) is the dilation factor of the current layer, determining the spacing between filters, and wi represents the size of the convolution kernel. Dilation introduces a fixed stride between adjacent filters, enabling the convolution operation to cover a wider range of input probabilities. By increasing the dilation factor, the receptive field of the network is expanded, allowing the TCN to capture local dependencies within the word-level probability sequences.

To further enhance the ATDPS model's ability to capture complex patterns and prevent gradient vanishing or exploding, we incorporate residual connections into the TCN architecture, following the approach proposed by [39]. These connections allow for the preservation of information across layers and can be expressed as:

$$H(w) = ReLu(F(w) + W_w w),$$

where d denotes the output after applying the residual connection, k is the rectified linear unit activation function, H(w) is a learnable weight matrix that ensures the input and output dimensions are consistent. By stacking multiple layers of these TCN blocks, a TCN-based encoder, denoted as ReLu, can be constructed.

Each word in the input text Ww is initially represented by its corresponding basic features. These basic features, derived from the word-level log probability lists, are fed into the TCN-based encoder to produce latent representations for each word. The encoder, through its hierarchical structure, captures local dependencies, transforming the basic features into contextualized latent representations. Mathematically, the output of the TCN-based encoder for the TCN (·)-th word S is given by:

$$x_i = TCN(l_i),$$

where i denotes the latent representation of wi from basic feature xi. This transformation enables ATDPS to effectively capture local dependencies of the input basic features, providing a more informative representation for subsequent processing.

After obtaining the probabilistic features wi and semantic features li, ATDPS concatenates these features into a single integrated representation. This concatenation process can be formalized as:

$$Z = [x_1 \oplus F_s, x_2 \oplus F_s, \ldots, x_n \oplus F_s],$$

where Fp=[x1,x2,...,xn] denotes concatenation operation, and Fs represents the integrated feature set for each word, resulting from concatenating the TCN-extracted probabilistic representation ⊕ with the corresponding semantic features Z.

3.5 Linear Classification Layer

The integrated feature set xi is processed through a self-attention-based context network designed to capture long-range dependencies by leveraging self-attention layers. This network enhances the ATDPS model's ability to encode contextualized representations across the word sequence, effectively incorporating global dependencies and refining interactions within the input. Specifically, the context network processes the integrated features to produce a sequence of contextualized representations, Fs, formally defined as:

$$C = Transformer(Z),$$

where Z represents the contextualized feature sequence obtained from two Transformer layers with fixed positional embeddings. Following the extraction of contextualized features, a linear classifier is trained to map each word's contextualized representation to a corresponding label. The prediction for each word C=[c1,c2,...,cn] is generated by:

$$y_i = softmax(Wc_i + b),$$

where C represents the predicted label for word wi, and yi and wi are learnable parameters of the classifier. To determine the sentence-level category, we count the predicted labels for each word and select the label with the highest frequency as the final label for the sentence. To optimize the model parameters, we use cross-entropy loss as the loss function, which is formally defined as:

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(y_i) + (1-y_i) log(1-y_i)],$$

where W denotes the ground truth label for word b.

4 Experiments

4.1 Datasets

To construct a comprehensive and diverse dataset for AI-generated text detection, we followed a methodology aligned with Wang et al. [15]. Our dataset consists of both human-written and AI-generated texts, derived from multiple domains. For the human-written portion, we selected documents from publicly available sources in the SnifferBench benchmark [19], including news articles (XSum [40]), web content [41], and social media posts [42]. To ensure content diversity and semantic richness, we randomly selected the first three sentences from each document to serve as prompts for LLM-based text generation. To produce AI-generated content, we employed a range of representative LLMs, including GPT-2, GPT-J, GPT-Neo, LLaMA-2, GPT-3.5-turbo, and GPT-4. Each model was provided with a consistent input template to promote coherence and comparability across outputs. Specifically, the prompt used was: *Please provide a continuation for the following content to make it coherent: {prompt}*. This template guided the LLMs to generate fluent and semantically relevant continuations, simulating realistic AI-generated responses to human-like writing.

To maintain the naturalness and authenticity of the generated text, we did not perform additional data preprocessing steps. This decision was made deliberately to better simulate real-world usage, where AI-generated and human-written texts appear in unprocessed form across applications. All samples were automatically labeled according to their source: LLM-generated texts were labeled as AI-generated, while texts from SnifferBench were labeled as human-written. This labeling process was deterministic and required no manual annotation.

The resulting dataset consists of 1,400 documents, with 200 documents per model category, totaling 28,802 sentences. Each category (human and AI-generated) is evenly represented. Following the same practice as in [15], we applied a stratified random split, allocating 90% of the data for training and 10% for testing, ensuring balanced distribution across all categories. To account for variance introduced by random sampling, we repeated the train-test split and model training five times using different random seeds and reported the average performance metrics across all runs.

To validate the linguistic coherence and appropriateness of the generated texts, we calculated the average perplexity of texts from each source using GPT-2 as the scoring model. Perplexity measures how well a language model predicts a given sequence; lower values suggest higher fluency and coherence. As shown in Table 2, GPT-3 and GPT-2 produce the lowest perplexity values, indicating smoother and more predictable outputs. GPT-4 and LLaMA-2 show moderate perplexity, while human-written texts exhibit the highest values--consistent with prior findings that human texts are more diverse and harder for language models to predict [14].

Table 2 Average perplexity of texts from different sources.

Text Source	Average Perplexity
GPT-2	8.73
GPT-Neo	11.39
GPT-J	11.86
LLaMA-2	16.14
GPT-3	8.45
GPT-4	13.31
Human	24.42

To demonstrate the generalization of ATDPS, we also employed a Multiclass AIGT Detection (MAD) benchmark dataset, introduced by Shi et al. [43]. This publicly available multiclass dataset comprises 10,608 text samples from human sources and seven prominent LLMs. It spans two domains reflecting real-world applications such as academic integrity and misinformation detection. We adopted the same data splitting strategy as in the original work.

4.2 Baseline Methods

Seven state-of-the-art baseline methods are employed for comparison:

BERT [38]: The BERT model is used as a feature extractor to extract semantic and syntactic features from the input text. The pre-trained BERT model is fine-tuned through supervised training on datasets of AI-generated and human-written texts.

RoBERTa [31]: RoBERTa is a pre-trained language model proposed by Facebook AI Research, based on the BERT architecture with several optimizations. It utilizes a larger training dataset and dynamic masking techniques to enhance the model's performance and robustness. RoBERTa is widely used in tasks such as text classification, question-answering systems, and text generation, and it has demonstrated excellent performance in multiple benchmark tests, making it an important tool in the NLP field.

SeqXGPT [15]: This method differentiates AIGT by leveraging probability features. Specifically, it uses language models such as GPT-2 to calculate the perplexity of each sentence and applies a manually selected threshold as the discriminative boundary, determining if each sentence is human-written or AI-generated.

DetectGPT [14]: DetectGPT uses their proposed z-score to differentiate AIGT. Multiple perturbations are added for each sentence (40 perturbations are tried for each sentence). Then, the z-score for each sentence is calculated based on a specific model. Again, a threshold is manually chosen to differentiate AI-generated sentences.

Sniffer [19]: The tool is widely used in fields such as public opinion monitoring, market research, and content recommendation, helping users gain deeper insights into text data and obtain valuable information. Through automated analysis, the text sniffer enhances the efficiency and accuracy of information processing, providing strong support for academic research and commercial applications.

AI-Catcher [44]: AI-Catcher is a multimodal deep learning model that integrates multilayer perceptron for linguistic and statistical feature representation with CNN for extracting sequential patterns from textual content. It fuses these representations to detect ChatGPT-generated scientific text, demonstrating high accuracy across human-written, AI-generated, and mixed content in diverse domains.

LECMiD [45]: This model combines RoBERTa-base embeddings with linguistically motivated features, such as lexical diversity and stylometry. It employs an autoencoder for classification and focuses on capturing fundamental differences between AI-generated and human-written texts, demonstrating strong generalization across domains and models.

4.3 Settings

For semantic features, ATDPS utilizes the BERT-base-uncased model[2]. For TCN, the input channel count is set to 1, and the network comprises five layers with channel dimensions of 64, 128, 128, 128, and 64, respectively. The hidden layer dimension is configured to 1024, and the number of heads in the multi-head attention mechanism is 12. We use the ReLU activation function and the AdamW optimizer, with hyperparameters set to yi of (0.9, 0.98). The learning rate is set to wi, weight decay is set to 0.1, and the batch size is set to 32. The hyperparameters for the baseline methods are kept consistent with the original papers.

For dataset configuration, we randomly partition the dataset by selecting 10% as the test set, leaving 90% for training. This sampling strategy is performed independently for each category to ensure balanced representation. Due to the randomness in selection and partitioning, we report the average results from five runs.

4.4 Experimental Results

In this section, we present a comprehensive analysis of the experimental results obtained from our study. Our primary objective is to validate the effectiveness of the ATDPS model in identifying AIGT by comparing it against seven state-of-the-art baseline models.

Table 3 Performance comparison of the ATDPS model and baseline models for AIGT detection on our dataset. Bold values indicate the best results, while underlined values represent the second-best results.

Model	Accuracy	F1	Recall	Precision
BERT	54.7	55.3	57.7	56.7
RoBERTa	54.5	56.7	55.1	57.5
SeqXGPT	83.6	83.5	82.6	83.6
DetectGPT	80.9	64.3	61.3	80.9
Sniffer	86.0	83.2	81.7	86.0
AI-Catcher	68.8	69.3	70.4	68.9
LECMiD	73.5	72.9	71.3	74.4
ATDPS	97.9	97.8	96.5	97.7

Overall Model Comparison. Table 3 summarizes the performance metrics--accuracy, precision, recall, and macro-average F1 score--of the ATDPS model in comparison with the selected baseline models on our dataset. The results indicate that the ATDPS model significantly outperforms all baseline models across all evaluated metrics. Specifically, ATDPS achieved an accuracy of 97.9%, a macro-average F1 score of 97.8%, a recall rate of 96.5%, and a precision of 97.7%. These results demonstrate the model's capability in effectively distinguishing AIGT texts, highlighting its advanced feature extraction and semantic understanding capabilities. To further verify the reliability of these improvements, we performed statistical significance testing based on five independent experimental runs. A two-tailed t-test comparing ATDPS and the best-performing baseline, Sniffer, confirmed that the differences in both accuracy and F1 score are statistically significant, with p-values below 0.01. This confirms that the observed performance gain is not attributable to random variation but reflects a consistent and robust advantage of the proposed approach.

In contrast, the performances of the baseline models, specifically BERT and RoBERTa, are markedly lower. For instance, BERT achieved an accuracy of only 54.7%, while RoBERTa fared slightly better at 54.5%. Such results underscore the inherent challenges that traditional models encounter when addressing the complexities of contemporary AIGT tasks. The relatively low performance of these models indicates limitations in their ability to adapt to the nuances and

intricacies of text generated by AI [27]. Furthermore, models such as SeqXGPT and Sniffer exhibited improved performance, with accuracy rates of 83.6% and 86.0%, respectively. However, even these models lag behind the performance of the ATDPS model. This performance gap illustrates the effectiveness of ATDPS in integrating probabilistic and semantic features through its use of a TCN architecture. By effectively encoding the probabilistic features, the ATDPS model captures both local and long-range dependencies, which are critical for accurate AIGT detection. Similarly, AI-Catcher and LECMiD achieved accuracies of 68.8% and 73.5%, respectively, which are superior to BERT and RoBERTa but still fall short of the probabilistic methods like SeqXGPT and Sniffer.

Table 4 Detection accuracy of the ATDPS model and baseline models across different generative models on our dataset. Bold values indicate the best results, while underlined values represent the second-best results.

Method	GPT-2	GPT-Neo	GPT-J	LLaMA-2	GPT-3	GPT-4	Human
BERT	44.1	39.0	42.3	50.0	65.4	50.7	77.7
RoBERTa	40.1	41.2	51.2	65.3	73.8	54.3	88.1
SeqXGPT	95.5	94.1	91.5	85.6	92.7	89.0	82.5
DetectGPT	76.5	77.1	82.0	85.3	88.1	79.0	83.2
Sniffer	85.1	84.4	83.8	84.3	86.2	85.5	86.9
AI-Catcher	72.5	68.3	70.4	65.5	69.8	64.3	73.2
LECMiD	81.2	79.7	77.4	73.2	72.0	64.6	87.1
ATDPS	100	100	100	92.0	94.7	88.9	100

Table 4 illustrates the detection accuracy of the ATDPS model and baseline models across various generative models on our dataset, demonstrating its superiority with a detection accuracy of 100% for both GPT-2 and GPT-Neo. This performance showcases the model's outstanding generalization ability and adaptability in recognizing AIGT. SeqXGPT also performs well across multiple models, achieving accuracy rates of 95.5% and 94.1% for GPT-2 and GPT-Neo, respectively. However, its accuracy on GPT-4, reported at 89.0%, indicates a slight decline, which may suggest limitations when handling next-generation generative models. AI-Catcher and LECMiD exhibit moderate performance across all categories, achieving accuracy between 64.3% and 81.2%. Their results suggest that linguistic or multimodal features alone do not sufficiently capture the subtle statistical patterns required for highly reliable AIGT detection.

By comparison, the performance of BERT and RoBERTa is generally low, particularly failing to compete effectively with newer models like GPT-2 and GPT-Neo. This suggests that the lack of probability features inherent in these traditional models can significantly hinder their performance in identifying the sources of AI-generated texts. Overall, the data presented reinforces the necessity for more sophisticated approaches in AIGT detection to overcome the limitations exhibited by conventional models.

The high accuracy on our dataset likely arises from its relative simplicity and high separability of features, where probabilistic and semantic cues exhibit clear distinctions between AI-generated and human-written texts--potentially due to domain-specific biases, limited variability in AI samples, or artifacts from the synthesis process. This aligns with findings by Wang et al. [15], who attribute high baseline performance to shortages in human-generated learning samples and context length limitations that restrict samples to only two distinct sources. Similarly, Huang et al. observe that high accuracy in in-domain settings often stems from training-test similarity, leading to brittleness under perturbations [46]. These high scores imply that the dataset may not fully capture the complexities of evolving LLMs, risking inflated metrics and reduced generalizability.

Table 5 Performance comparison of the ATDPS model and baseline models for AIGT detection on the MAD dataset. Bold values indicate the best results, while underlined values represent the second-best results.

Model	Accuracy	F1	Recall	Precision
BERT	72.6	72.4	72.6	72.3
RoBERTa	73.2	72.9	73.2	72.8
SeqXGPT	95.7	95.6	95.7	96.1
Sniffer	91.2	92.3	91.2	92.4
AI-Catcher	74.5	74.8	74.5	76.0
LECMiD	79.4	79.7	79.4	80.4
ATDPS	98.4	97.6	98.4	97.7

Table 5 presents the performance metrics on the MAD dataset. ATDPS achieves an accuracy of 98.4%, and a macro-average F1 score of 97.6%, outperforming all baselines. SeqXGPT follows closely with 95.7% accuracy, while Sniffer attains 91.2%. The linguistic and multimodal baselines, LECMiD and AI-Catcher, yield accuracies of 79.4% and 74.5%, respectively, surpassing BERT and RoBERTa but lagging behind our models. These findings confirm that integrating TCN-encoded probabilistic features with semantic representations improves the model's robustness across diverse text distributions.

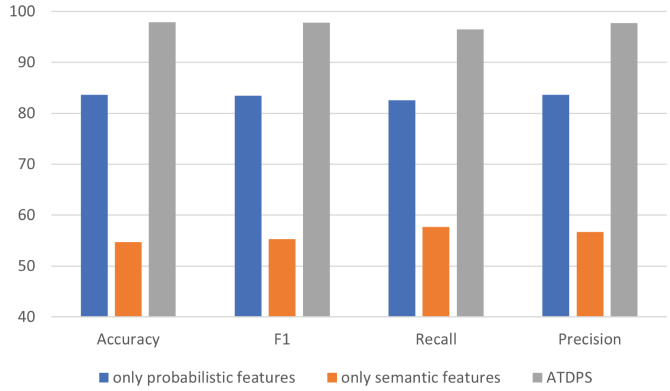

Figure 2 *Ablation Study.*

Ablation Study. The ablation experiment illustrated in Figure 2 provides an analysis of how different feature combinations affect the performance of the ATDPS model. Specifically, ``only probabilistic features'' refers to the model's performance when utilizing only probabilistic features encoded through TCNs, while ``only semantic features'' pertains to the model's performance using semantic features for sentence-level classification. The results clearly demonstrate that the full ATDPS model, which integrates both feature types, consistently outperforms all other configurations across all evaluation metrics.

When using only probabilistic features, the model achieves relatively strong performance, particularly in terms of accuracy and recall--both exceeding 80%. This underscores the substantial discriminative power of probabilistic features in distinguishing between AI-generated and human-written text. On the other hand, the semantic-only configuration performs notably worse. This degradation is consistent with prior findings by Li et al. [19], who show that semantic-only models often exhibit low accuracy when the available training data is limited. Semantic representations generally require large datasets to learn class-distinguishing features; moreover, high-quality LLM outputs exhibit strong semantic coherence, reducing the separability of AI-generated and human text in semantic embedding space.

Despite their limited standalone efficacy, semantic features provide substantial improvements when fused with probabilistic features. As reported in [47] and corroborated by our experiments, semantic features capture the deep expressions of text from the perspectives of overall content and writing style. Probabilistic features are sensitive to local lexical anomalies, whereas semantic features are attuned to overall stylistic consistency; combining the two mitigates the limitations inherent in a single-perspective approach. This fusion leverages their complementary strengths: probabilistic features detect local statistical irregularities, while semantic features capture global stylistic coherence.

In summary, the ablation study in Figure 2 highlights the importance of feature integration in AI-generated text detection. By jointly leveraging probabilistic and semantic features, the ATDPS model achieves superior and balanced performance, demonstrating the necessity of a multifaceted approach in addressing the nuanced challenges of this task.

5 Conclusions

In this study, we propose a novel AI-generated Text Detection model combining Probabilistic and Semantic features (ATDPS) to address the challenge of detecting AIGT. Our model leverages the strengths of TCNs to process the probabilistic features extracted from multiple LLMs. By integrating these probabilistic features with the semantic features derived by BERT, we construct a robust feature set that captures the intricate nuances of both human-written and AI-generated texts. The integration of these complementary features is facilitated through a Transformer-based architecture, which enables our model to discern long-range dependencies and contextual relationships within the text. Extensive comparative experiments demonstrate the superior performance of the ATDPS model across various evaluation metrics. The results indicate that our model significantly outperforms existing state-of-the-art methods, underscoring its efficacy in identifying AIGT amidst the rapidly evolving landscape of LLMs.

To promote broader applicability, we constructed a diverse dataset comprising news articles, social media posts, and web content, and included texts generated by a wide range of LLMs from GPT-2 to GPT-4. This design strengthens the model's generalization across domains and generative styles. Nonetheless, the current implementation focuses solely on English texts. Extending the model to multilingual or low-resource settings remains an important direction for future work.

Although ATDPS exhibits strong performance under standard conditions, it has several limitations. Its reliance on white-box access to an ensemble of LLMs for extracting probabilistic features imposes notable practical constraints. Furthermore, the computational overhead of querying multiple large LLMs for each text sample is prohibitive, potentially increasing inference time and resource demands by several factors compared to single-model or black-box alternatives. In future work, we plan to explore hybrid approaches that reduce white-box dependency, such as approximating probabilistic features using surrogate models or employing zero-shot black-box methods that leverage perplexity-based perturbations or token-level entropy analysis without requiring full model access.

References

1. Yang Z, Zhang Y, Sui D, Ju Y, Zhao J, Liu K. Explanation guided knowledge distillation for pre-trained language model compression. ACM Trans Asian Low Resour Lang

Inf Process. 2024;23(2):1-19. 10.1145/3639364

2. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: Scaling language modeling with pathways. Journal of Machine Learning Research. 2023;24(240):1-113. 10.48550/arXiv.2204.02311

3. Wu J, Yang S, Zhan R, Yuan Y, Chao LS, Wong DF. A survey on LLM-generated text detection: Necessity, methods, and future directions. Comput Linguist. 2025;51(1):275-338. 10.1162/coli_a_00549

4. Li Y, Li Q, Cui L, Bi W, Wang Z, Wang L, et al. MAGE: Machine-generated text detection in the wild. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 36-53. 10.18653/v1/2024.acl-long.3

5. Gao W, Li L, Zhu X, Wang Y. Detecting disaster-related tweets via multimodal adversarial neural network. IEEE Multimed. 2020;27(4):28-37. 10.1109/mmul.2020.3012675

6. Miah MSU, Kabir MM, Sarwar TB, Safran M, Alfarhood S, Mridha M. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM. Sci Rep. 2024;14(1):1-18. 10.1038/s41598-024-60210-7

7. Gao W, Zheng C, Zhu X, Deng H, Wang Y, Hu G. Knowledge-injected prompt learning for actionable information extraction from crisis-related tweets. Comput Electr Eng. 2024;118:1-11. 10.1016/j.compeleceng.2024.109398

8. Feng S, Wan H, Wang N, Tan Z, Luo M, Tsvetkov Y. What does the bot say? Opportunities and risks of large language models in social media bot detection. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 3580-3601. 10.18653/v1/2024.acl-long.196

9. Gao W, Deng H, Zhu X, Fang Y. Topic-BERT: Detecting harmful information from social media. Intell Decis Technol. 2021;15(3):333-342. 10.3233/idt-200094

10. Gao W, Ni M, Deng H, Zhu X, Zeng P, Hu X. Few-shot fake news detection via prompt-based tuning. J Intell Fuzzy Syst. 2023;44(6):9933-9942. 10.3233/jifs-221647

11. Wang Y, Mansurov J, Ivanov P, Su J, Shelmanov A, Tsvigun A, et al. M4GT-Bench: Evaluation benchmark for black-box machine-generated text detection. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 3964-3992. 10.18653/v1/2024.acl-long.218

12. Heumann M, Kraschewski T, Breitner MH. ChatGPT and GPTZero in research and social media: A sentiment- and topic-based Analysis. In: Proceedings of the Americas Conference on Information Systems (AMCIS). 2023. p. 1-10. 10.2139/ssrn.4467646

13. Gehrmann S, Strobelt H, Rush AM. GLTR: Statistical detection and visualization of generated text. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2019. p. 111-116. 10.18653/v1/p19-3019

14. Mitchell E, Lee Y, Khazatsky A, Manning CD, Finn C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In: Proceedings of the International Conference on Machine Learning (ICML). 2023. p. 24950-24962. 10.48550/arXiv.2301.11305

15. Wang P, Li L, Ren K, Jiang B, Zhang D, Qiu X. SeqXGPT: Sentence-Level AI-Generated Text Detection. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP). 2023. p. 1144-1156. 10.18653/v1/2023.emnlp-main.73

16. Li S, Sun C, Xu Z, Tiwari P, Liu B, Gupta D, et al. Toward explainable dialogue system using two-stage response generation. ACM Trans Asian Low Resour Lang Inf Process. 2023;22(3):1-18. 10.1145/3551869

17. Dixit A, Kaur N, Kingra S. Review of audio deepfake detection techniques: Issues and prospects. Expert Syst. 2023;40(8):1-19. 10.1111/exsy.13322

18. Shakarami A, Nicole L, Terreran M, Tos APD, Ghidoni S. TCNN: A transformer convolutional neural network for artifact classification in whole slide images. Biomed Signal Process Control. 2023;84:1-19. 10.1016/j.bspc.2023.104812

19. Li L, Wang P, Ren K, Sun T, Qiu X. Origin tracing and detecting of LLMs. arXiv [preprint]. 2023. 10.48550/arXiv.2304.14072

20. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-Shot learners. In: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS). 2020. p. 1-25. 10.5555/3495724.3495883

21. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and efficient foundation language models. arXiv [preprint]. 2023. 10.48550/arXiv.2302.13971

22. Bruera A, Tao Y, Anderson A, Cokal D, Haber J, Poesio M. Modeling brain representations of words' concreteness in context using GPT-2 and human ratings. Cogn Sci. 2023;47(12):1-48. 10.1111/cogs.13388

23. Zellers R, Holtzman A, Rashkin H, Bisk Y, Farhadi A, Roesner F, et al. Defending against neural fake news. In: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS). 2019. p. 9051-9062. 10.48550/arXiv.1905.12616

24. Adelani DI, Mai H, Fang F, Nguyen HH, Yamagishi J, Echizen I. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In: Proceedings of the International Conference on Advanced Information Networking and Applications (AINA). 2020. p. 1341-1354. 10.1007/978-3-030-44041-1_114

25. Lund BD, Wang T, Mannuru NR, Nie B, Shimray S, Wang Z. ChatGPT and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. Journal of the Association for Information Science and Technology.

2023;74(5):570-581. 10.2139/ssrn.4389887

26. Kim ZM, Lee KH, Zhu P, Raheja V, Kang D. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 5449-5474. 10.18653/v1/2024.acl-long.298

27. Jawahar G, Abdul-Mageed M, Lakshmanan LVS. Automatic detection of machine generated text: A critical survey. In: Proceedings of the International Conference on Computational Linguistics (COLING). 2020. p. 2296-2309. 10.18653/v1/2020.coling-main.208

28. Bakhtin A, Gross S, Ott M, Deng Y, Ranzato M, Szlam A. Real or fake? Learning to discriminate machine from human generated text. arXiv [preprint]. 2019. 10.48550/arXiv.1906.03351

29. Fagni T, Falchi F, Gambini M, Martella A, Tesconi M. TweepFake: About detecting deepfake tweets. arXiv [preprint]. 2020. 10.1371/journal.pone.0251415

30. Solaiman I, Brundage M, Clark J, Askell A, Herbert-Voss A, Wu J, et al. Release strategies and the social impacts of language models. arXiv [preprint]. 2019. 10.48550/arXiv.1908.09203

31. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A robustly optimized bert pretraining approach. arXiv [preprint]. 2019. 10.48550/arXiv.1907.11692

32. Ippolito D, Duckworth D, Callison-Burch C, Eck D. Automatic detection of generated text is easiest when humans are fooled. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2020. p. 1808-1822. 10.18653/v1/2020.acl-main.164

33. Sadasivan VS, Soltanolkotabi M, Feizi S. CUDA: Convolution-based unlearnable datasets. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). 2023. p. 3862-3871. 10.1109/cvpr52729.2023.00376

34. Kumar A, Levine A, Goldstein T, Feizi S. Certifying model accuracy under distribution shifts. arXiv [preprint]. 2022. 10.48550/arXiv.2201.12440

35. Kirchenbauer J, Geiping J, Wen Y, Katz J, Miers I, Goldstein T. A watermark for large language models. In: Proceedings of the International Conference on Machine Learning (ICML). 2023. p. 17061-17084. 10.48550/arXiv.2301.10226

36. Hou AB, Zhang J, Wang Y, Khashabi D, He T. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 1706-1715. 10.18653/v1/2024.findings-acl.98

37. Krishna K, Song Y, Karpinska M, Wieting J, Iyyer M. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS). 2023. p. 1-32. 10.52202/075280-1195

38. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). 2019. p. 4171-4186. 10.18653/v1/N19-1423

39. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). 2016. p. 770-778. 10.1109/cvpr.2016.90

40. Narayan S, Cohen SB, Lapata M. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP). 2018. p. 1797-1807. 10.18653/v1/d18-1206

41. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019;1(8):1-9. Available from: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf 10.1007/979-8-8688-0515-8_1

42. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word bectors for sentiment analysis. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). 2011. p. 142-150. Available from: https://aclanthology.org/P11-1015/ 10.3115/1073416.1073420

43. Shi Y, Sheng Q, Cao J, Mi H, Hu B, Wang D. Ten words only still help: Improving black-Box AI-generated text detection via proxy-guided efficient re-sampling. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). 2024. p. 494-502. 10.24963/ijcai.2024/55

44. Alhijawi B, Jarrar R, AbuAlRub A, Bader A. Deep learning detection method for large language models-generated scientific content. Neural Comput Appl. 2025;37(1):91-104. 10.1007/s00521-024-10538-y

45. Petukhova K, Kazakov R, Kochmar E. PetKaz at SemEval-2024 task 8: Can linguistics capture the specifics of LLM-generated text?. In: Proceedings of the International Workshop on Semantic Evaluation (SemEval). 2024. p. 1140-1147. 10.18653/v1/2024.semeval-1.166

46. Huang G, Zhang Y, Li Z, You Y, Wang M, Yang Z. Are AI-Generated Text Detectors Robust to Adversarial Perturbations?. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 2024. p. 6005-6024. 10.18653/v1/2024.acl-long.327

47. Shimada H, Kimura M. A method for distinguishing model generated text and human written text. J Adv Inf Technol. 2024;15(6):714-722. 10.12720/jait.15.6.714-722