IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 February 2024, accepted 7 March 2024, date of publication 13 March 2024, date of current version 21 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3376693

RESEARCH ARTICLE

DeepTextMark: A Deep Learning-Driven Text Watermarking Approach for Identifying Large Language Model Generated Text

TRAVIS MUNYER, ABDULLAH ALL TANVIR, ARJON DAS, AND XIN ZHONG
Department of Computer Science, University of Nebraska Omaha, Omaha, NE 68182, USA
Corresponding author: Xin Zhong (xzhong@unomaha.edu)

ABSTRACT The rapid advancement of Large Language Models (LLMs) has significantly enhanced the capabilities of text generators. With the potential for misuse escalating, the importance of discerning whether texts are human-authored or generated by LLMs has become paramount. Several preceding studies have ventured to address this challenge by employing binary classifiers to differentiate between human-written and LLM-generated text. Nevertheless, the reliability of these classifiers has been subject to question. Given that consequential decisions may hinge on the outcome of such classification, it is imperative that text source detection is of high caliber. In light of this, the present paper introduces DeepTextMark, a deep learning-driven text watermarking methodology devised for text source identification. By leveraging Word2Vec and Sentence Encoding for watermark insertion, alongside a transformer-based classifier for watermark detection, DeepTextMark epitomizes a blend of blindness, robustness, imperceptibility, and reliability. As elaborated within the paper, these attributes are crucial for universal text source detection, with a particular emphasis in this paper on text produced by LLMs. DeepTextMark offers a viable ''add-on'' solution to prevailing text generation frameworks, requiring no direct access or alterations to the underlying text generation mechanism. Experimental evaluations underscore the high imperceptibility, elevated detection accuracy, augmented robustness, reliability, and swift execution of DeepTextMark.

INDEX TERMS Text source detection, large language model text detection, text watermarking, deep learning.

I. INTRODUCTION

Large Language Models (LLMs), such as ChatGPT [1], have recently achieved notable success. The advancements in LLMs can be advantageous across various domains, yet there also lies the potential for inappropriate applications. A prevailing concern regarding publicly accessible LLMs is the challenge in distinguishing between machine-generated and human-written text, a difficulty that persists even in instances of misuse [2]. For instance, students might utilize automatically generated texts as their own submissions for assignments, evading conventional ''plagiarism'' detection. The high fidelity of the text generated by LLMs exacerbates the challenge of detection, marking a significant hurdle. Again, there exist advanced text augmentation methods capable of effortlessly modifying any given text [3], [4] [5], [6]. Therefore, devising a method to ascertain the origin of text could serve as a valuable approach to curtail similar misapplications of LLMs.

Various classifiers have been developed to differentiate between LLM-generated text and human-written text [2], [7]. However, the efficacy of these classifiers remains somewhat constrained at present. Numerous studies have explored the accuracy of these classifiers [8], along with techniques to circumvent classifier detection [9]. A reliable source detection mechanism that is challenging to bypass is crucial, given its potential applications in identifying plagiarism and misuse. Therefore, employing text watermarking for text

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

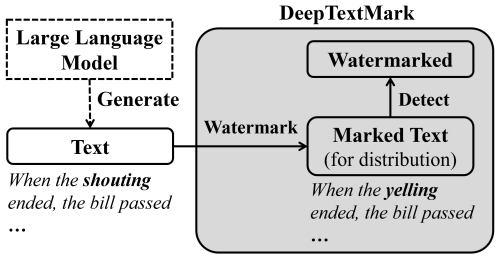


FIGURE 1. Overall idea of DeepTextMark.

source detection appears to be a prudent approach, as it is both reliable and challenging to circumvent.

Text watermarking entails the covert embedding of information (*i.e.*, the watermark) into cover texts, such that the watermark is only discernible by authorized detectors. While watermarking is more conventionally applied to images [10], its application to text can enable the identification of text originating from specific sources, such as an LLM (refer to figure 1 for the proposed source detection mechanism). However, conventional text watermarking techniques often necessitate manual intervention by linguists, exhibit a lack of robustness, and do not possess the blindness property. Specifically, these traditional techniques are prone to minor modifications of the watermarked text (lacking robustness), and necessitate the original text for the extraction or detection of the watermark (lacking blindness). For a watermarking technique to be practically viable in detecting LLM-generated text, the method should be scalable (*i.e.*, automatic). Moreover, since the watermark detector may not have access to the original text at the time of detection, it should not require it (*i.e.*, blind). Additionally, the detection process should be highly reliable, aiming to achieve superior classification accuracy. Ideally, the watermarked text should remain imperceptible, ensuring the natural preservation of the text's meaning. Lastly, the classification mechanism should be resilient to minor alterations of the text (*i.e.*, robust).

A nascent method has been proposed for embedding watermarks into LLMs [11]. However, a notable limitation of this method is its requisite access to the text generation phase of the LLMs, a requirement that may not be practical in real-world applications, particularly when the source models of the LLMs are not open-source. This dependency poses a significant challenge as many LLMs are proprietary or their internals are not publicly disclosed, thereby restricting the applicability of such watermarking techniques. Moreover, without the requisite access to the text generation phase, implementing watermark-based source detection mechanisms becomes inherently challenging. This highlights the necessity for developing alternative watermarking techniques that are both effective and adaptable to varying levels of access to the LLMs' internal workings.

This paper introduces DeepTextMark, a robust and blind deep learning-based text watermarking method principally aimed at detecting LLM-generated text. DeepTextMark employs word substitution, utilizing a pre-trained amalgam of Universal Sentence Encoder embeddings [12] and Word2Vec [13] to identify semantically congruent substitution words. The inserted watermark is invisible to the naked eye, and the alterations made to the text, such as substituting words with synonyms while keeping the grammatical structure intact, are designed to ensure that the watermark remains undetectable to readers. Therefore, it preserves the imperceptible nature of the watermark within the text. Moreover, we propose a novel classifier, grounded in transformer architecture [14], to discern watermarked text, enhancing detection accuracy and robustness. This classifier can accurately differentiate between marked and unmarked sentences based solely on the content and features extracted from the text, without altering its appearance or readability in any noticeable way. This imperceptibility ensures that the watermark remains covert and undetectable to human observers, thereby preserving its effectiveness for authentication or tracking purposes without alerting potential infringers to its existence. This amalgam of pre-trained models for substitution word selection and the transformer-based watermark detector underscore the novel contributions of this paper. Being deep learning-driven, the watermarking and detection techniques are scalable and fully automatic. The classifier necessitates only the watermarked text for highly accurate classification, epitomizing the technique's blindness. Furthermore, the paper elucidates an extension of this technique to multiple sentences, like essays, accentuating a primary application. Empirical evidence is provided demonstrating near-perfect accuracy as text length increases, enriching the method's reliability, especially with a modest sentence count.

The primary contributions encapsulate: (1) an "add-on" text watermarking method for detecting generated text without necessitating access to the LLMs' generation phase; (2) an automatic and imperceptible watermark insertion method; and (3) a robust, high-accuracy deep learning-based text watermark detection method, rendering DeepTextMark a valuable asset in the realm of text authenticity verification.

The rest of this paper is organized as follows. We discuss related works in section II. The watermark insertion and detection process is discussed in section III. Experiments showing the reliability, imperceptibility, robustness, and empirical runtime are shown in section IV followed by a conclusion of the work in section V.

II. RELATED WORK

Our contributions are summarized as robust detection of LLM-generated text, a novel method for text watermarking insertion, and a novel approach for text watermarking detection; the following sections provide a review of related work in these domains. Section II-A offers a concise review of state-of-the-art methods for LLM-generated text detection, while Section II-B delves into classical text watermarking techniques.

A. TEXT SOURCE DETECTION FOR LARGE LANGUAGE MODELS

Recent endeavors have been directed towards developing classifiers aimed at differentiating between LLM-generated text and human-written text. The prevailing approach entails the collection and labeling of LLM-generated and human-written texts, followed by the training of a binary classifier through supervised learning. Although the efficacy of these classifiers has yet to be fully established, some preliminary analyses have been reported [8], [9]. One study [9] elucidated three distinct methods, substantiated with examples, to circumvent the GPTZero [7] classifier detection. Another investigation [8] conducted a direct assessment of GPTZero's accuracy, uncovering inconsistencies in its ability to detect human-written text. Moreover, classifier-based LLM-generated text detectors commonly necessitate a substantial character count to perform detection accurately. For instance, GPTZero [7] required a minimum of 250 characters to initiate detection. Looking ahead, OpenAI is planning a cryptography-based watermarking system for ChatGPT-generated text detection [15], although no definitive work has been disclosed as of yet. Zero-shot learning-based methods have also demonstrated some advancement. For example, Mitchell et al. [16] reported an increment in AUROC from 1% to 14% compared to other zero-shot detection strategies across various datasets; however, the accuracy might still fall short in real-world applications concerning text generated by models.

A method has been proposed for detecting LLM-generated texts based on text watermarking [11], which involves watermarking the text by modifying the LLMs (sensitive tokens are defined and excluded from the output of the LLMs). In contrast, our proposed DeepTextMark does not necessitate access to or modifications of the LLM. Distinct from model-dependent methods, DeepTextMark exhibits a model-independent feature, enabling its application to any text. Moreover, DeepTextMark employs a substantially more compact architecture with about 50 million parameters, whereas the method in [11] necessitates billions of parameters to implement the watermarking process.

A pertinent topic in text watermarking for identifying generated text is the potential use of paraphrasing attacks to bypass AI-detectors, as elaborated in a study by Sadasivan et al. [17]. This concern is not relevant to our target scenario, as DeepTextMark focuses solely on the detection of text output by an LLM. Should a human writer meticulously rewrite the text generated by an LLM, the resultant paraphrased text may not be subject to ''plagiarism'' detection in our scenario.

Relative to existing state-of-the-art methods, our proposal exhibits several advantages: (1) Our watermarking method renders detection bypass challenging unless the LLM-generated text is rewritten, as the watermark is embedded in undisclosed locations, necessitating a rewrite for its removal. Once rewritten, the text is deemed as distinct human-written text; (2) The method demonstrates high detection accuracy, nearing 100%, which significantly elevates with an increasing number of sentences, substantiated through binomial distribution analysis. Even on a single sentence, a reliable detection rate of 86.52% is achieved; (3) To our knowledge, this is the inaugural LLM-independent, deep learning-based general text watermarking method; (4) Unlike some methods necessitating access to text generation processes, our approach requires no access to the LLM's original text generation, allowing our watermarker to function as an ''add-on'' to the LLM system (see Figure 1).

B. TRADITIONAL TEXT WATERMARKING

Common classical text watermarking methods can be categorized into open space, syntactic, semantic, linguistic, and structural techniques. A brief summary of each of these techniques is provided below.

1) OPEN SPACE

The open space method embeds a watermark into text data by adding extra white space characters or spaces at specific locations in the text [18]. For instance, extra white space between words or lines could be encoded as a 1, while normal white space could encode as a 0. The strategy for adding extra white space and its encoding is subject to the implementation. Although the open space method can be simple to implement and automate, it may be susceptible to watermark removal without altering the text's meaning, as an individual could easily eliminate the extra white space.

2) SYNTACTIC

Certain word orders can be altered without changing the meaning or grammatical correctness of a sentence. The syntactic method watermarks text by modifying the order of words in sentences [19]. For example, ''this and that'' could encode to 1, and ''that and this'' could encode to 0. However, this method may not scale well since many sentences do not have sequences of words that qualify for reordering. Additionally, this method might necessitate manual intervention by a linguist, as developing an automated system to detect reorderable words could be challenging.

3) SEMANTIC

Semantic text watermarking techniques embed the watermark by substituting words with synonyms [19]. While the semantic method can be automated, as briefly discussed in this paper, classical implementation requires the original text to detect the watermark (i.e., classical semantic text watermarking is non-blind). Moreover, determining which word to replace, and selecting an appropriate synonym, presents a non-trivial challenge.

4) LINGUISTIC

The linguistic category of text watermarking amalgamates semantic and syntactic techniques, embedding watermarks

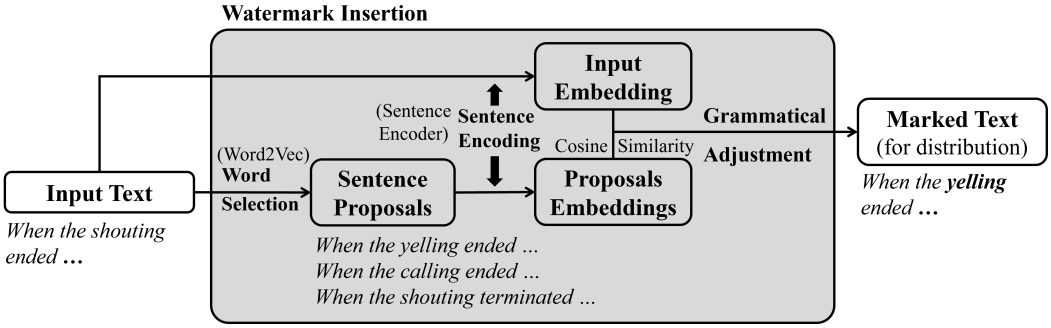


FIGURE 2. Watermark insertion details.

into text through a blend of word rearrangement and synonym replacement [19].

5) STRUCTURAL

The structural technique replaces certain symbols with visually similar letters and punctuation, albeit with different Unicode representations [20]. It may be relatively straightforward to detect these symbols either manually due to minor visual differences, or automatically by identifying characters from uncommon Unicode sets. Reverting the watermarking without altering the text's meaning could also be straightforward. Due to these limitations, structural techniques do not align with our primary objective of watermarking text generated by LLMs.

Contrastingly, we employ word2vec [13] and the Universal Sentence Encoder [12] for watermark insertion, and devise a transformer-based model for watermark detection. This approach aligns well with our target application of text source detection, as it facilitates blindness while enhancing imperceptibility, robustness, and reliability. Our watermark insertion and detection methodology is rooted in deep learning, distinguishing our method from traditional text watermarking techniques.

III. THE PROPOSED DEEPTEXTMARK

This section presents the details of DeepTextMark. The proposed watermark insertion and detection schemes are respectively discussed in Sections III-A and III-B. This discussion shows the automatic and blindness traits achieved by DeepTextMark. Section III-C analyzes the application scenario of DeepTextMark to multiple sentences.

A. WATERMARK INSERTION

In contemporary settings, individuals employ extensive language models to produce textual content and subsequently rephrase it using synonymous words as a strategy to circumvent plagiarism. This serves as the rationale behind our introduced watermark insertion model, aiming to detect alterations in text even when someone attempts to paraphrase content generated by large language models in order to evade plagiarism detection. The watermark insertion process is presented in Figure 2.

1) WORD SELECTION

Given a sentence, we initially segregate candidate words from punctuation, stopwords [21], and whitespace, preserving these elements to retain the original sentence structure. Each candidate word is then transposed to an embedding vector utilizing a pre-trained Word2Vec model [13]. A roster of replacement words is engendered by identifying the n nearest vectors to the candidate word vector in the Word2Vec embedding space, where n is a pre-defined integer, and reconverting these vectors back into words. We engender a list of sentence proposals by substituting each candidate word with its list of replacement words, thereby fabricating unique sentence variations. The loci of the watermark in each sentence proposal are indirectly ascertained by Word2Vec. Each unique variation is deemed a sentence proposal, representing a potential watermarked sentence. Empirically, employing a larger corpus of nearest vectors allows for the consideration of an augmented set of replacement words and consequently more sentence proposals, potentially ameliorating imperceptibility albeit at the expense of elevated processing time. We also delved into various word-level watermarking techniques. Initially, a sole word within each sentence was substituted with its synonyms which we denote as **single word synonym substitution**. This scope was subsequently broadened to encompass multiple-word replacements within each sentence which is denoted by **multiple word synonyms substitution**. In the terminal phase of our experimentation, we embraced a flexible approach, permitting the substitution of any candidate word with an available synonym in a sentence.

2) SENTENCE ENCODING

At this juncture, each sentence proposal is evaluated solely based on word-level quality. We ascertain that the quality of the watermarked sentence is enhanced when the architecture is allowed to consider sentence-level quality. To facilitate this, we employ a pretrained Universal Sentence Encoder [12] to score the quality of each sentence proposal.

TABLE 1. Example sentence candidates of correct and incorrect synonym selections.

1. The September-October **term** jury had been charged by Fulton Superior Court judge Durwood Pye to investigate reports of possible "irregularities" in the hard-fought primary which was won by mayor-nominate Ivan Allen Jr.
2. The September-October **terms** jury had been charged by Fulton Superior Court judge Durwood Pye to investigate reports of possible "irregularities" in the hard-fought primary which was won by mayor-nominate Ivan Allen Jr.
3. The September-October **condition** jury had been charged by Fulton Superior Court judge Durwood Pye to investigate reports of possible "irregularities" in the hard-fought primary which was won by mayor-nominate Ivan Allen Jr.

TABLE 2. Example sentence candidates with varied synonyms.

The **primary** thing she did was to take off her hat and then as she had no other covering she.
The **first** thing she did was to take off her hat and then as she had no other covering she.
The **leading** thing she did was to take off her hat and then as she had no other covering she.

This encoder transposes a sentence into a high-dimensional vector representation. Initially, both the original sentence and each sentence proposal are transposed into vector representations using the Universal Sentence Encoder. Subsequently, we compute the similarity score for each sentence proposal by measuring the cosine similarity between the vector representation of the original sentence and that of the sentence proposal. The sentence proposal exhibiting the highest similarity score is identified as the potential watermarked sentence. Given that the watermarking process necessitates no human intervention, the methodology is rendered automatic.

3) GRAMMATICAL ADJUSTMENT

In pursuit of mitigating grammatical inaccuracies, essential measures have been undertaken. Our methodology encompasses word substitution with synonymous counterparts, whilst steadfastly preserving the original sentence structure. In this vein, we have eschewed the elimination of stopwords or the alteration of punctuation, thereby safeguarding sentence integrity.

The process of synonym selection is meticulously designed to favor optimal replacements. Nevertheless, challenges emerge in instances where the most apt synonym diverges in grammatical structure or meaning. For instance, replacing the term 'elections,' a plural noun, with 'election,' its singular counterpart, could engender grammatical incongruity. To forestall such scenarios, a preliminary determination of the grammatical number of the target word is initiated with a class engine [22] which employs diverse methods to facilitate plural and singular inflections, the selection of ''a'' or ''an'' for English words based on pronunciation, and the manipulation of numbers represented as words. This module comprehensively provides plural forms for nouns, most verbs, and select adjectives, including ''classical'' variants like transforming ''brother'' to ''brethren'' or ''dogma'' to ''dogmata.'' Singular forms of nouns are also available, allowing the choice of gender for singular pronouns, such as transforming ''they'' to ''it,'' ''she,'' ''he,'' or ''they.'' Pronunciation-based ''a'' or ''an'' selection is extended to all English words and most initialisms. It is crucial to note that when using plural inflection methods, the word to be inflected should be the first argument, expecting the singular form; passing a plural form may yield undefined and likely incorrect results. Similarly, the singular inflection method anticipates the plural form of the word. The plural inflection methods also offer an optional second argument indicating the grammatical ''number'' of the word or another word for agreement. Subsequently, synonyms congruent with the grammatical form of the original word are curated.

Algorithm 1 Watermark Insertion

```
function WatermarkInsertion(input_text)
    word_embedder ← Word2Vec
    sentence_encoder ← SentenceEncoder
    input_embeddings ← Encode(word_embedder, input_text)
    sentence_proposals ← GenerateProposals(input_text)
    proposals_embeddings ← Encode(sentence_encoder, sentence_proposals)
    best_proposal ← ComputeCosineSimilarity(input_embeddings, proposals_embeddings)
    marked_text ← GrammaticalAdjustment(best_proposal)
    return marked_text
```

A few examples of sentence candidates with correct and incorrect synonym selections are presented in Table 1. It is imperative to note that when we scrutinize the word **term**, we encounter the closest synonyms, some of which contravene the grammatical criteria due to their distinct grammatical numbers, with one being singular and the other plural. Consequently, given that our initial word is in the singular form, our consideration is limited exclusively to synonyms in the singular form. Consequently, in lieu of employing **terms**, we opt to substitute it with **condition**.

Analogous complexities arise concerning parts of speech, as certain words harbor synonyms across diverse lexical categories. To adeptly navigate this intricacy, integration of the classic POS (Part of Speech) tagger [23] has been effected. Post identification of the word's grammatical number, the endeavor to pinpoint synonyms aligning with its specific part of speech is undertaken. This bifurcated approach underpins both syntactic and grammatical consistency in our synonym substitution process.

A few examples of sentence candidates with varied synonyms selections are presented in Table 2. An analysis of the term **primary** reveals that the closest synonyms are typically adverbs like **first**, thereby deviating from the grammatical

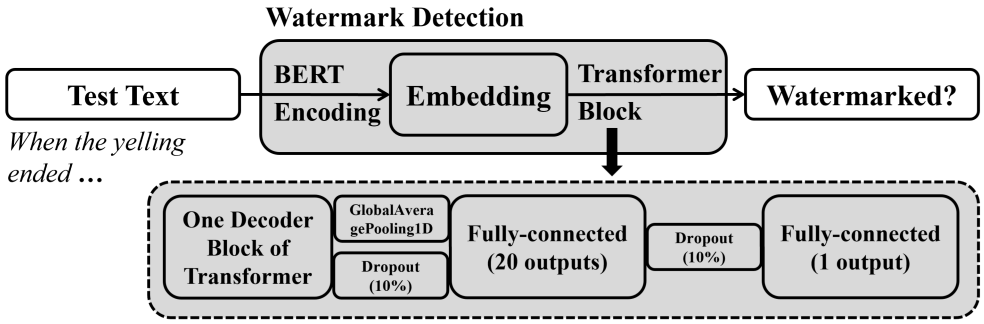


FIGURE 3. Watermark detection details.

condition, as the original term is a proper noun (NNP). Given the categorical distinction that our original word falls into the proper noun category (NNP), our focus is exclusively on synonyms that share this grammatical property. This rationale informs our decision to replace the word **primary** with **leading** instead of **first**. We have implemented the type of parts of speech by using the POS-tagger provided by the NLTK [23]. Specifically, we employed the Penn Treebank POS tagger. The tagging process involved tokenization of input text, breaking it into individual words or sentences, and subsequently assigning part-of-speech tags to each word. The POS tagging was conducted using a Hidden Markov Model (HMM), trained on a large annotated corpus, such as the Penn Treebank corpus, wherein the model learned the probabilities of transitions between different POS tags and the probabilities of observing specific words given a certain POS tag. The Viterbi algorithm was employed during the tagging of new text to identify the most likely sequence of POS tags given the observed words and the learned probabilities. This approach proved effective for obtaining accurate and contextually relevant part-of-speech annotations in diverse textual datasets. Algorithm 1 outlines the entire operational process.

B. WATERMARK DETECTION

The watermark detector operates as a binary classifier categorizing inputs into ''watermarked'' and ''unmarked'' classes, leveraging network architectures inherent in transformers [14]. We have used the Bidirectional Encoder Representations from Transformers (BERT) pre-trained model which is capable of capturing the contextual meaning of words in a sentence. Unlike traditional methods that treat each word as independent, BERT considers the entire context of the sentence, including the relationships between words. Hence, it will possess the capability to recognize sentence modifications and distinguish between marked and unmarked sentences. Furthermore, BERT serves as a powerful feature extractor, automatically extracting high-dimensional representations of text at various levels of granularity. It's scalability and generalization capabilities enable it to handle diverse datasets and adapt to different domains and languages with minimal additional training. The architecture of this classifier is delineated in Figure 3.

The watermark detection classifier endeavors to minimize the ensuing binary cross-entropy loss:

$$\mathcal{L} = y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)), \qquad (1)$$

where y_i denotes the label, and $p(y_i)$ represents the predicted probability. The parameters of the BERT encoder are initially frozen, allowing the loss to converge with the transformer block being trainable. Upon convergence of the loss with a frozen BERT, the parameters of BERT are unfrozen, the learning rate of Adam is attenuated, and training is recommenced until loss convergence is reattained. This iterative training paradigm can precipitate a notable enhancement in prediction performance training solely the transformer block. The outcomes of the training regimen are elaborated in section IV-B. This architecture, post convergence, embodies the watermark detector. Given that the detector necessitates no access to the original data for prediction execution, the methodology is characterized as blind.

C. WATERMARK DETECTION FOR MULTIPLE SENTENCES

A prominent application scenario for the proposed watermarking technique is its deployment on a collection of sentences. Consequently, the classification outcome is contingent on the majority classification rendered for each individual sentence. Employing the binomial distribution, it can be demonstrated that the likelihood of accurately classifying a sentence collection converges to near perfection as the volume of sentences in the collection escalates, provided the probability of accurately classifying a single sentence is reasonably high ($> \ 85\%$). Notwithstanding, a superior probability of correct classification for a single sentence implies a reduced sentence count is requisite to attain near-perfect accuracy. Algorithm 2 comprehensively outlines the entire working procedure.

The proof underpinning this claim is articulated as follows: Presume the probability of accurately classifying a sentence as watermarked or not is denoted by p, and remains consistent across all sentences. In a scenario where at least half of the sentences in a text comprising n sentences are accurately

Algorithm 2 Watermark Detection

```
function WatermarkDetection(test_text)
    embeddings ← BERT_Encode(test_text)
    decoder_output ← TransformerDecoderBlock(embeddings)
    pooled_output ← Pooling(decoder_output)
    dropout_output ← Dropout(pooled_output)
    fc_20_output ← FullyConnected(dropout_output, 20)
    dropout_fc ← Dropout(fc_20_output)
    watermark_score ← FullyConnected(dropout_fc, 1)
    return watermark_score
```

classified, the entire text is deemed correctly classified. It can be substantiated that the probability of accurately classifying exactly x sentences can be encapsulated by the binomial probability, denoted as $P(x)$. Hence, the probability $P(x > \lceil n/2 \rceil)$ can be formulated as the summation in Equation (2):

$$P(x > \lceil n/2 \rceil) = \sum_{i=\lceil n/2 \rceil}^{n} \binom{n}{i} \times p^i \times (1-p)^{n-i}. \quad (2)$$

IV. EXPERIMENTS

This section illustrates the effectiveness of DeepTextMark by analyzing its properties in regard of text watermarking. Dataset preparation is explained in section IV-A. The reliability of the watermark detection is shown in section IV-B. Section IV-C explains the ablation study. Section IV-D provides a summary of the imperceptibility, and the imperceptibility and detection accuracy trade-off. Comparisons are made between DeepTextMark and traditional text watermarking methods. Section IV-E provides an analysis of the experiments used to test robustness, which is followed by an evaluation of the empirically observed runtime in section IV-G.

A. DATASET

1) TRAINING DATA

A dataset comprising 34,489 sentences was assembled from the Dolly ChatGPT Dataset [24]. This approach aims to underscore the generalization capability of the proposed DeepTextMark. Robust performance across diverse textual genres exemplifies the model's aptitude for generalizing to arbitrary text. Evaluations have been also conducted on texts engendered by expansive language models such as ChatGPT, as depicted in an instance in Figure 4. Within the training set, half of the sentences are watermarked employing the methodology delineated in section III-A, whilst the remainder are retained unaltered. This yields a dataset encompassing nearly 17,000 watermarked samples and approximately 17,000 unmarked samples. The corpus of watermarked and unmarked sentence samples are randomly amalgamated, with 75% earmarked for training, and the residual 25% allocated for validation—this composition underpins the training of the detector. To facilitate the assessment of imperceptibility in section IV-D, a dataset encapsulating all 34,489 sentences as original and watermarked pairs is retained.

TABLE 3. Sentence count on detection accuracy (%) (single synonym).

Num Sentences	Dolly (%)	C4 (%)
1	86.52	76.30
5	98.02	90.97
10	99.92	98.49
20	99.99	99.75
30	100.00	99.96
50	100.00	99.99
60	100.00	100.00

TABLE 4. Sentence count on detection accuracy(%) (multiple synonyms).

Num Sentences	Dolly (%)	C4 (%)
1	94.87	95.72
5	99.88	99.92
10	100.00	100.00

TABLE 5. Ablation study.

Experiment	A	B	C	D
Single Synonyms	✓	✓		
Multiple Synonyms			✓	✓
Handling Singular/Plural Number		✓		✓
Handling Parts of Speech		✓		✓
Detection Accuracy (%)	88.36	86.52	92.74	94.87

2) TESTING DATA

We assessed the performance of our model by subjecting it to testing using C4 datasets [25] containing multiple sentences. To evaluate its performance, we systematically extracted 100 tokens at a time, aggregating them into a unified dataset featuring numerous sentences. This process yielded a total of 8,800 datasets. Subsequently, we conducted rigorous testing on these datasets, incorporating both single and multiple synonym substitutions to gauge the model's adaptability and effectiveness.

B. WATERMARK DETECTION ACCURACY

The proposed watermark detection classifier is trained using the dataset discussed in Section III-B. We train the architecture with the parameters of the pre-trained BERT encoder frozen for 6 epochs, with an Adam learning rate set to 0.0001. Then, we unfreeze the pre-trained BERT architecture, reduce the learning rate of Adam [26] to 0.000001, and train for 50 more epochs. In our training model, 148 million parameters have been used. The result validation accuracy, which represents the sentence-level detection accuracy on the dolly validation dataset, is 86.52% for single synonyms and 94.87% for multiple synonyms. And for C4 datasets, its 76.30% for single synonyms and 95.72% for multiple synonyms substitution.

Additionally, we conduct this training process on several versions of the dataset, each with an increasing number of

```
"deep learning, a subset of machine learning, has
emerged as a subverter epitome in artificial
intelligence, mimicking the man brain's neural
meshing architecture to operation vast sum of
information and selection meaningful patterns. at
its core, trench encyclopedism relies on
artificial neural net with multiple bed (deep
neural networks) to progressively learn and
represent intricate hierarchy of characteristic
from altogether stimulation data. this attack has
proven remarkably successful across diverse
domains, from look-alike and delivery credit to
cancel nomenclature processing. the might of
trench encyclopedism prevarication in its power
to automatically discover and learn intricate
representations, enabling the growth of
sophisticated manikin capable of tackling
composite tasks. while its achiever are evident,
challenge such as interpretability, information
dependency, and computational essential persist,
underscoring the indigence for ongoing enquiry to
unlock the full potency of this transformative
technology."
```

FIGURE 4. A watermarked example from ChatGPT with prompt "Give me a short essay about deep learning".

sentences. We observe that as we continually increase the size of the dataset, the validation accuracy improves. Training with an increasing number of sentences could further improve the sentence-level prediction accuracy. We find that the current training is balanced on table 3, table 4 and Section IV-D, as this validation yields near-perfect prediction accuracy with only a small collection of sentences.

As elucidated by the binomial distribution in Section III-C, the probability of accurately classifying a collection of sentences markedly increases with the augmentation of the sentence count in the text, attributable to our sentence-level insertion process. Assuming the likelihood of accurately classifying a single sentence aligns closely with the validation accuracy computed during training, and that this likelihood remains consistent across all sentences, we can forecast the probability of accurately classifying a collection of sentences utilizing the summation outlined in Eq. (2). Under this assumption, the probability of correct prediction corresponding to varying sentence counts is tabulated in Table 3 and Table 4. Table 3 and Table 4 underscore the reliability of the method, highlighting an increased likelihood of accurate detection as the number of sentences rises. This trend is observed for both single and multiple synonyms substitution, encompassing both dolly and C4 datasets.

C. ABLATION STUDY

To evaluate the effectiveness of our proposed method, we conducted an ablation study by systematically removing components from our model and observing their impact on performance. Specifically, we conducted four experiments denoted as A, B, C, and D, each representing a variant of our model with varying degrees of complexity. Experiment D, which incorporates all proposed enhancements, achieved the highest accuracy among the tested configurations. This result

TABLE 6. A few example sentences: 1. the original text; 2. watermarked text by the traditional method; 3. watermarked text by the DeepTextMark. with single synonym substitution; and 4. watermarked text by the DeepTextMark with multiple synonyms substitution.

1. Which episodes of season four of game of thrones did michelle maclaren direct.
2. Which **installment** of season four of game of thrones did michelle maclaren direct.
3. Which **sequence** of season four of game of thrones did michelle maclaren direct.
4. Which **sequence** of season **quaternity** of game of thrones did michelle maclaren direct.
1. Who saved andromeda from the sea monster?
2. Who saved andromeda from the **ocean** monster?
3. Who saved andromeda from the **ocean** monster?
4. Who saved andromeda from the **ocean monstrosity**?

suggests that the additional components in experiment D contribute positively to the overall performance of the model. Furthermore, by comparing the accuracy of experiment D with those of experiments A, B, and C, as shown in Table 5, we can pinpoint the specific contributions of each component to the model's effectiveness. Our findings underscore the importance of the incorporated enhancements and highlight the significance of their inclusion in our proposed approach.

D. IMPERCEPTIBILITY OF WATERMARK INSERTION

A sentence bearing an imperceptible watermark should maintain grammatical correctness and retain the same meaning as the original sentence. Thus, the imperceptibility of text watermarking should be gauged by sentence meaning similarity. The Universal Sentence Encoder [12] encapsulates the semantic meaning of sentences into an embedding vector, enabling the measurement of sentence meaning similarity through the computation of cosine similarity between two sentence embeddings. Hence, we propose to quantify the imperceptibility of text watermarking using the Sentence Meaning Similarity (SMS):

$$\text{SMS} = S(\text{encode}(o), \text{encode}(m)), \tag{3}$$

where o denotes the original text, m denotes the watermarked text, encode$(\cdot)$ represents a neural network that computes a semantic embedding (e.g., the Universal Sentence Encoder), and S is a function that computes the similarity between the vectors (cosine similarity is utilized in this paper). Computing the mean SMS (mSMS) over a dataset provides an average measure of text watermark imperceptibility. We have performed our experiment for the test dataset discussed in Section III-B and we are able to achieve 0.9765 mSMS for single synonyms and 0.9892 *mSMS* for multiple synonyms while the traditional method provides 0.9794 *mSMS*. The high mSMS value exemplifies the imperceptible watermarking of texts by DeepTextMark. An illustration of watermarking a text produced by ChatGPT is presented in Figure 4, with additional examples of original and watermarked paragraphs available in the supplementary documents.

TABLE 7. Remove sentences attack.

Removed Sentences	Watermarked *mIOA*	Unmarked *mIOA*
1	0.000	0.022
3	0.000	0.033
5	0.001	0.028
10	0.029	0.030
15	0.058	0.169
17	0.110	0.191
19	0.206	0.280

TABLE 8. Add sentences attack.

Added Sentences	Watermarked *mIOA*	Unmarked *mIOA*
1	0.000	0.026
3	0.001	0.061
5	0.003	0.115
10	0.069	0.269

For analytical purposes, we implement traditional text watermarking and test the proposed watermark detection network at a single-sentence level. Specifically, we create an implementation of semantic watermarking using WordNet [27] to select synonyms. Although this traditional method achieved some success, the replacement word occasionally rendered the sentence nonsensical, as this method did not account for sentence structure. Some sentence examples are illustrated in Table 6 (additional sentence examples can be found in the supplementary documents). While the traditional method can be effectively detected by our detection network for a single sentence, as depicted in Table 10, the *mSMS* on the collected dataset significantly improves with DeepTextMark multiple synonyms and not that much far from single synonyms as well.

E. ROBUSTNESS

Robustness in the domain of image watermarking implies that the watermark must remain invariant to malicious attacks or unintentional modifications [28]. Translating this notion of robustness to text watermarking is fairly straightforward. A robust text watermarking method should ensure that removing the watermark is challenging, whether the removal attempts are unintentional, arising from normal processing, or intentional attacks targeting the watermark. For watermark detection to fail, the watermarked text should need to be altered beyond recognition.

Given that this is an emerging area, no standard method exists to measure robustness for text watermarking [19]. Therefore, we propose a metric named Mean Impact of Attack (*mIOA*) to measure robustness. The *IOA* is defined as follows:

$$IOA(x, y) = (1-|detect(x)-y|)-(1-|detect(x_a)-y|), \quad (4)$$

where x represents the target data (text of one or more sentences in this paper), x_a denotes the attacked data obtained by arbitrarily attacking x, y signifies the label for x (watermarked or unmarked), and $detect(\cdot)$ denotes the utilization of the detection network to output the predicted label of the input. *IOA* gauges the change in accuracy following an attack on the data. A positive *IOA* indicates a detrimental effect on prediction performance due to the attack, while a negative *IOA* indicates improved prediction performance post-attack (which should be rare). An *IOA* further from 0 (either less than or greater than) signifies a higher impact from the attack. An *IOA* of 0 indicates the attack did not affect the prediction accuracy. Calculating the mean *IOA* over a dataset yields the *mIOA*.

TABLE 9. Replace sentences attack.

Replaced Sentences	Watermarked *mIOA*	Unmarked *mIOA*
1	0.001	0.013
3	0.008	0.058
5	0.040	0.126
7	0.103	0.210

TABLE 10. Comparative analysis in terms of *mSMS* and detection accuracy.

Method	*mSMS*	Detection Accuracy
DeepTextMark (Single Synonyms)	0.9765	0.8652
DeepTextMark (Multiple Synonyms)	0.9892	0.9487
Traditional	0.9794	0.8836

1) DATA FOR ROBUSTNESS TEST

We have prepared two sets of data: one with watermarked text and one with unmarked text. Each set contains 1000 collections, with each collection comprising 20 sentences. These sentences are randomly selected from the testing set described in Section III-B.

We then define several attacks and compute the *mIOA* for each attack to gauge the robustness of our watermarking technique. These attacks are designed to progressively modify the text, with the severity of each attack increasing the dissimilarity between the modified and original texts. Each attack also represents a common interaction with the text. By attacking both watermarked and unmarked data, we aim to evaluate the detection accuracy for both types of data, which helps ensure that our system is equally effective at detecting watermarks and identifying unmarked data.

2) REMOVE SENTENCES ATTACK

We remove a selected number of n sentences from the text. This action reduces the watermark presence in the text, thus challenging the robustness of the detection. Table 7 presents the *mIOA* on both watermarked and unmarked datasets for several values of n. In all cases, the total number of sentences is 20.

The results show that the *mIOA* increases as the severity of the attack intensifies (i.e., more sentences are removed), yet the performance remains commendable as the *mIOA* stays close to 0. Interestingly, the *mIOA* is consistently higher on the watermarked data.

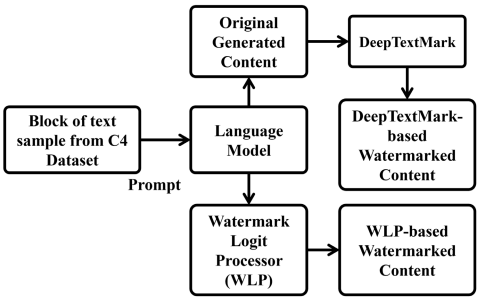


FIGURE 5. Sample generation process for testing and comparing watermark detection accuracy of DeepTextMark and WLP.

TABLE 11. Detection accuracy of DeepTextMark and WLP with smaller datasets.

Model	Detection Accuracy (%)
WLP	92.43
DeepTextMark	90.74

3) ADD SENTENCES ATTACK

In this attack, a specified number of sentences (represented by n) with the opposite label are randomly added to the text. For instance, watermarked sentences are added to unmarked text. Increasing the value of n challenges the robustness of the detection, as it dilutes the percentage of text that corresponds to the expected label. Table 8 illustrates the *mIOA* on the watermarked and unmarked datasets for several values of n.

The data shows that the *mIOA* increases as n increases. DeepTextMark maintains a high performance, as the *mIOA* remains close to 0 for a reasonable n.

4) REPLACE SENTENCES ATTACK

This attack adopts a similar data dilution approach as the add sentences attack. It distorts the text data by replacing n existing sentences in the text with randomly selected sentences of the opposite type, where n is a specified integer. Table 9 presents the watermarked and unmarked *mIOA* for several values of n.

The *mIOA* increases as n increases, and DeepTextMark remains close to 0, indicating a minimal impact on detection performance. Since the modified text becomes increasingly dissimilar to the original text post-attack, an escalating performance impact is expected and acceptable as the severity of each attack intensifies. These experiments affirm that DeepTextMark is robust to text modifications stemming from common text interactions.

F. COMPARATIVE ANALYSIS

To start our comparative analysis, we have used three methods and their combinations. The first approach is the Traditional method, involving a simple single-word modification without any components of the proposed DeepTextMark. Following that, we introduce DeepTextMark, which encompasses both single and multiple synonym substitutions while rectifying grammatical errors. We perform the comparison

TABLE 12. Robustness comparison of DeepTextMark and WLP.

Model	ϵ	TPR	FNR
multinomial sampling	0.1	0.819	0.181
multinomial sampling	0.3	0.353	0.647
multinomial sampling	0.5	0.094	0.906
multinomial sampling	0.7	0.039	0.961
beam search	0.1	0.834	0.166
beam search	0.3	0.652	0.348
beam search	0.5	0.464	0.536
beam search	0.7	0.299	0.701
DeepTextMark	-	0.830	0.170

in terms of imperceptibility and detection accuracy which has been shown in table 10 where we can see that our DeepTextMark with multiple synonyms performed very well in terms of both imperceptibility and detection accuracy.

We have also performed a deeper comparative analysis between our approach and the method proposed by Kirchenbauer et al. [11]. For clarity within the context of this paper, we will refer to their method as the Watermark Logit Processor (WLP) method, to prevent any naming confusion. It's important to highlight that the WLP method necessitates access to LLMs, specifically utilizing them as a logit processor to favor the selection of ''green'' tokens during text generation. On the other hand, our proposed method operates independently and does not require access to LLMs.

To ensure a fair comparison, it is imperative that both methods are evaluated using the same source of text, specifically an LLM. Consequently, for text generation, we have employed the Open Pre-trained Transformer (OPT-2.7B). The primary objective of this experiment is to apply our method, alongside the WLP method, to watermark the content generated by OPT-2.7B and subsequently evaluate the detection accuracy for comparison purposes. To generate a substantial amount of text content, we utilized a subset of the C4 dataset, comprising 22k text samples, as the source of prompts for the LLM in a seeded environment to yield deterministic outputs with a set of 500 sequences of length T = 200 token sequences which is similar to the WLP paper. The authors of WLP papers proposed two different methods which we denote **WLP-multinomial sampling** and **WLP-beam search** to avoid confusion. With this setup, upon inputting text (prompt) samples from the C4 dataset into the base LLM, we obtain blocks of text, which we term the ''Original Generated Content.'' Subsequently, we apply our proposed method to the ''Original Generated Content'' to produce the DeepTextMark-based watermarked content. Conversely, when we incorporate the WLP logit processor with the base LLM, the identical input text samples yield the WLP Watermarked Content. Figure 5 illustrates the text generation methodology employed for the comparative evaluation between DeepTextMark and WLP. In this configuration, our model demonstrates a notable detection rate of 90.66%. This outcome, achieved despite training on the distinct Dolly dataset, underlines the robust generalization capability of our approach, affirming its effectiveness across diverse datasets.

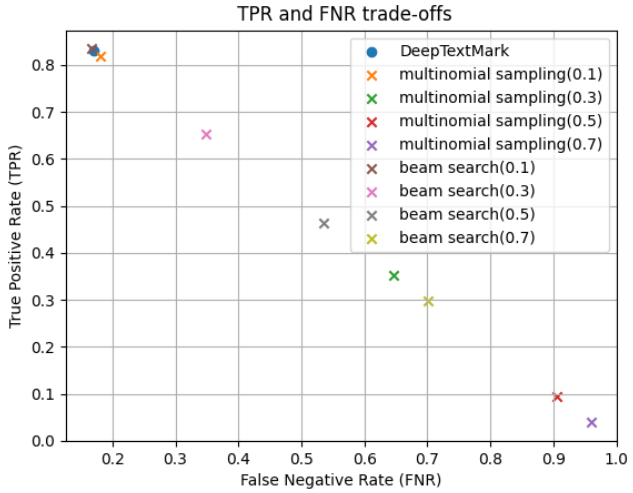


FIGURE 6. TPR and FNR trade-offs.

TABLE 13. DeepTextMark Runtime on a single CPU core.

Component	Time per Sentence (seconds)
Insertion	0.27931
Detection	0.00188

In our experimental evaluation, we utilized a subset of 500 data points to assess the watermark detection performance of both models. Despite the distinct datasets employed in training our model DeepTextMark, it demonstrates a commendable detection rate, only marginally lower than that reported in the WLP paper. Specifically, DeepTextMark achieved an accuracy of 90.74%, closely approaching the 92.42% accuracy of the WLP model. This proximity in performance is noteworthy, considering the differences in training datasets. Table 11 presents a detailed comparative analysis of the watermark detection accuracies between DeepTextMark and WLP.

We conducted a robustness comparison between the two models, considering three attack types: text insertion, deletion, and substitution. Text insertion attacks add extra tokens post-generation, while text deletion removes tokens from the generated output, potentially diminishing text quality by reducing the effective language model (LM) context width. Text substitution attacks involve replacing one token with another, which can be automated through dictionary or LM techniques but may degrade text quality.

Our comparative analysis, summarized in Table 12, reveals the robustness of DeepTextMark and WLP. The WLP study involved meticulous parameter adjustments to optimize their model's performance. Despite being trained on the Dolly Dataset, our model exhibited superior performance when tested on the C4 dataset produced by the LLM, outperforming in most scenarios for watermark detection accuracy. For robustness evaluation, we introduced new metrics while also using the True Positive Rate (TPR) and False Negative Rate (FNR) metrics from the WLP paper to ensure a fair assessment.

The Area Under the Receiver Operating Characteristic (AUC) curve and True Positive Rate (TPR) are key metrics in binary classification. AUC illustrates the trade-off between sensitivity (TPR) and 1 - specificity (False Positive Rate) across different thresholds, ranging from 0 to 1. A value of 0.5 implies no discriminative ability, whereas 1 indicates perfect classification. Higher AUC values denote superior model performance. TPR, or sensitivity/recall, is the ratio of correctly identified positive instances to all actual positives, defined as: $\text{TPR} = \mathit{TruePositives}/(\mathit{TruePositives} + \mathit{FalseNegatives})$. Conversely, the False Negative Rate (FNR) quantifies the proportion of positives incorrectly classified as negatives: $\text{FNR} = \mathit{FalseNegatives}/(\mathit{Positives} + \mathit{FalseNegatives})$

A superior TPR, signifying DeepTextMark's proficiency in correctly identifying positive instances while minimizing false negatives, underscores its efficacy in capturing the majority of actual positive cases. Concurrently, the smaller FNR suggests a reduced probability of overlooking positive instances, highlighting DeepTextMark's competence in averting false negatives and precisely identifying positive cases. In light of our model's outperformance compared to WLP, it can be inferred that DeepTextMark demonstrates a heightened capability in detecting watermarked sentences, surpassing the performance of WLP in this regard. This substantiates the conclusion that our model excels in discerning watermarked content more effectively.

Figure 6 delineates the interplay between TPR and FNR for our proposed method about the established method, WLP. Each data point on the plot encapsulates the performance of a method at distinct decision thresholds. The visual examination of the scatter plot underscores that our method does not lag behind WLP in terms of TPR and FNR characteristics across various operational points. This observation is crucial in establishing the efficacy of our method, aligning it favorably with the performance benchmarks set by WLP.

G. EMPERICAL RUNNING SPEED

This section evaluates the running speed of DeepTextMark. The experiments concerning running speed are conducted on an Intel i9-13900k CPU. We measure the time taken for watermark insertion across 1000 unmarked sentences and compute the sentence-level average watermark insertion time. Similarly, we time the watermark detection process on 1000 watermarked sentences and compute the average detection time. The average times for watermark insertion and detection, in seconds, are provided in Table 13.

As demonstrated, both the insertion and detection processes run quickly, serving as efficient "add-on" components for text source detection. The insertion process incurs a higher overhead compared to the detection process. It is important to note that these experiments were conducted using only a single core of the CPU. By parallelizing the implementation, the overhead from the insertion process could be significantly reduced, especially on server-level

machines, which are typically employed to implement LLMs in our target application scenario.

V. CONCLUSION

Recently, the use of LLMs has surged significantly in both industry and academia, mainly for text generation tasks. Nevertheless, in certain scenarios, it is crucial to ascertain the source of text—whether it is generated by an LLM or crafted by a human. Addressing this requirement, we introduce a deep learning-based watermarking technique designed for text source identification, which can seamlessly integrate with existing LLM-driven text generators. Our proposed method, DeepTextMark, stands out due to its blind, robust, reliable, automatic, and imperceptible characteristics. Unlike common direct classification techniques [7] for source detection that demand a substantial amount of characters for accurate prediction, our watermarking technique enables both watermark insertion and detection at the sentence level. Our findings demonstrate that with the insertion of watermarks, the accuracy of our detection classifier can approach near-perfection with merely a small set of sentences. Given that the watermark is embedded in each sentence individually, the robustness and reliability of the watermark enhance with an increasing number of sentences. The core advantages of our work include: an ''add-on'' text watermarking method facilitating the detection of generated text without requiring access to the LLMs' generation phase; an automatic and imperceptible method for watermark insertion; and a robust, high-accuracy, deep learning-based text watermark detection methodology.

While DeepTextMark introduces a significant advancement in text watermarking using deep learning, we recognize a few areas where future enhancements could be beneficial. First, the effectiveness of DeepTextMark is closely tied to the representativeness of the training data. Efforts to diversify this data could further improve its applicability across various text styles and languages. Second, as DeepTextMark functions in a 'plug-in' manner, its utility is contingent on the initial watermarking of the generated text. Without pre-watermarking, detection capabilities are limited, pointing to a dependency that may affect its applicability in certain scenarios. Lastly, while the method currently shows promising results in watermarking texts of standard lengths, we are exploring ways to adapt it more effectively for very short or stylistically diverse texts. These limitations represent opportunities for ongoing research and underscore the potential for continuous improvement in the field of AI-driven text watermarking.

In conclusion, our study has successfully introduced DeepTextMark, a novel deep learning-driven approach for text watermarking, offering a robust solution for distinguishing between human-authored texts and those generated by large language models. As we look toward the future, several promising directions can further enhance and expand the utility of our approach. We envision enhancing the robustness of DeepTextMark against more advanced text manipulation techniques, especially those using AI-based rewriting tools, to maintain its effectiveness in increasingly sophisticated digital environments. Moreover, exploring scalability to manage larger and more diverse datasets will be crucial in adapting our method for big data applications. Another significant direction involves extending the compatibility of DeepTextMark with various large language models, broadening its applicability across different AI-generated text scenarios. Developing real-time applications, such as content management system plugins, will also be pivotal in dynamically detecting and managing AI-generated content. Lastly, we acknowledge the importance of addressing the ethical and legal implications surrounding text watermarking, particularly in terms of privacy and data security in the age of AI. This aspect is critical to ensuring that our methodologies align with societal norms and legal standards. As we continue to build upon the foundation laid by DeepTextMark, these future endeavors will undoubtedly contribute to the evolving landscape of text watermarking and AI-generated content detection, reinforcing the importance of authenticity and integrity in digital communications.

REFERENCES

[1] (2023). *ChatGPT*. Accessed: Jul. 10, 2023. [Online]. Available: https://openai.com/blog/chatgpt

[2] (2023). *New AI Classifier for Indicating AI-Written Text*. Accessed: May 2, 2023. [Online]. Available: https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

[3] A. Onan, ''GTR-GA: Harnessing the power of graph-based neural networks and genetic algorithms for text augmentation,'' *Exp. Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120908.

[4] A. Onan and K. Filiz Balbal, ''Improving Turkish text sentiment classification through task-specific and universal transformations: An ensemble data augmentation approach,'' *IEEE Access*, vol. 12, pp. 4413–4458, 2024.

[5] A. Onan, ''SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 7, Jul. 2023, Art. no. 101611.

[6] A. Onan, ''Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2098–2117, May 2022.

[7] (2023). *GPTZero*. Accessed: Jul. 10, 2023. [Online]. Available: https://gptzero.me/

[8] (2023). *Is Gptzero Accurate? Can It Detect Chatgpt? Here's What Our Tests Revealed*. Accessed: Jul. 10, 2023. [Online]. Available: https://nerdschalk.com/is-gptzero-accurate-detect-chat-gpt-detector-tested/

[9] (2023). *Testing Gptzero: A Trending CHATGPT Detection Tool*. Accessed: Jul. 10, 2023. [Online]. Available: https://michaelsheinman.medium.com/testing-gptzero-a-trending-chatgpt-detection-tool-3ee14a056543

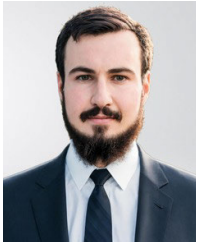
[10] H. Fang, Z. Jia, Z. Ma, E.-C. Chang, and W. Zhang, ''PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network,'' in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2267–2275.

[11] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, ''A watermark for large language models,'' in *Proc. 40th Int. Conf. Mach. Learn.*, vol. 202, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds. Jul. 2023, pp. 17061–17084.

[12] D. Cer, Y. Yang, S. Yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, ''Universal sentence encoder,'' 2018, *arXiv:1803.11175*.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010.
[15] (2023). *How the ChatGPT Watermark Works and Why It Could Be Defeated*. Accessed: Jul. 10, 2023. [Online]. Available: https://www.searchenginejournal.com/chatgpt-watermark/475366#close
[16] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-shot machine-generated text detection using probability curvature," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 24950–24962.
[17] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can AI-generated text be reliably detected?" 2023, *arXiv:2303.11156*.
[18] C. Ou, "Text watermarking for text document copyright protection," *Comput. Sci.*, vol. 725, Jun. 2003. [Online]. Available: https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/725ou.pdf
[19] N. Shamimi Kamaruddin, A. Kamsin, L. Yee Por, and H. Rahman, "A review of text watermarking: Theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.
[20] S. G. Rizzo, F. Bertini, and D. Montesi, "Fine-grain watermarking for intellectual property protection," *EURASIP J. Inf. Secur.*, vol. 2019, no. 1, p. 10, Jul. 2019, doi: 10.1186/s13635-019-0094-2.
[21] K. V. Ghag and K. Shah, "Comparative analysis of effect of stopwords removal on sentiment classification," in *Proc. Int. Conf. Comput., Commun. Control (IC)*, Sep. 2015, pp. 1–6.
[22] P. Dyson. *Inflect*. Accessed: Mar. 10, 2024. [Online]. Available: https://pypi.org/project/inflect/
[23] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python*. Sebastopol, CA, USA: O'Reilly Media, 2009. [Online]. Available: https://www.nltk.org/book/
[24] Databrickslab. (2023). *Dolly 15k Dataset*. [Online]. Available: https://github.com/databrickslabs/dolly/tree/master/data
[25] Hugging Face. *Hugging Face Datasets*. Accessed: Mar. 10, 2024. [Online]. Available: https://huggingface.co/datasets/c4
[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, 2015, pp. 1–15.
[27] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: 10.1145/219717.219748.
[28] W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun, "A comprehensive survey on robust image watermarking," *Neurocomputing*, vol. 488, pp. 226–247, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222002533



TRAVIS MUNYER received the B.S. degree in computer science and cybersecurity from the University of Nebraska Omaha, in May 2023. He is currently pursuing the M.S. degree in computer science from Georgia Institute of Technology, with a specialization in interactive intelligence.

From 2020 to 2023, he was an Undergraduate Researcher with the Machine Learning and Computer Vision Group, University of Nebraska Omaha. He is a Software Engineer with a well-known tech company headquartered in Olathe, KS, USA. His research interests include computer vision, natural language processing, image and text watermarking, and applications of machine learning to cybersecurity.

Mr. Munyer was a recipient of the Outstanding Cybersecurity Graduate Award from the University of Nebraska Omaha.

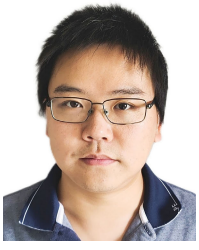
ABDULLAH ALL TANVIR is currently pursuing the Ph.D. degree with the University of Nebraska Omaha, in the research area of machine learning and computer vision.

Prior to the Ph.D. pursuits, he applied his expertise as a Machine Learning Engineer in a renowned IT company. In this role, he actively contributed to the development of innovative solutions and leveraging machine learning techniques to solve complex problems. His practical experience in industry has complemented his academic endeavors, providing him with a holistic perspective on the application of theoretical concepts in real-world scenarios. His research interests include artificial intelligence, machine learning, computer vision, and natural language processing.

Mr. Tanvir was awarded the GRACA Fund in recognition of his exceptional research achievements at the University of Nebraska Omaha.

ARJON DAS received the B.S. degree in computer science and engineering from Chittagong University of Engineering and Technology, Bangladesh, in 2018, and the M.S. degree in computer science from the University of Nebraska Omaha, Omaha, NE, USA, in 2023.

From 2021 to 2023, he was a Research Assistant with the RNA Laboratory, University of Nebraska Omaha. His research interests include computer vision, self-supervised learning, and image and text watermarking.



XIN ZHONG received the Ph.D. degree from New Jersey Institute of Technology, Newark, NJ, USA, in 2018. He is currently an Assistant Professor with the Department of Computer Science, University of Nebraska Omaha. His research interests include digital image processing and analysis, computer vision, pattern recognition, computational intelligence, machine learning, deep learning, and image watermarking.