Comparing Watermarking and Transformer Methods in Terms of Accuracy and Computational Efficiency

Ernest Osei Amoafo

March 27, 2026

Acknowledgements

I would like to express my profound gratitude to my supervisor, Jakob Dieterle, for his invaluable guidance, constructive feedback, and continuous support throughout the duration of this project. His insights significantly contributed to the refinement of the methodology and the depth of the final analysis. I also acknowledge the Mathematics Institute at Georg-August-Universität Göttingen for providing the academic environment and resources necessary to conduct this High-Performance Data Analytics (HPDA) research.

Contents

List of Figures

1 Abstract

Rapid advancement of Large Language models (LLMs) has made AI-generated content increasingly indistinguishable from human works and, as such, resulted in growing concerns about academic integrity, misinformation, content authenticity, and copyright issues. This report represents a comprehensive study on detecting AI-generated content, focusing on text. Drawing from recent relevant literature, watermarking techniques (proactive) and transformer-based methods (reactive) were reviewed.

A research question is proposed to compare their accuracy and computational efficiency. Using a simple implementation with PyTorch, a simple classifier is demonstrated to achieve higher accuracy on the dataset. Results highlight the strengths of transformer models like DistilBERT, achieving up to 97% accuracy in reviewed studies. Future research could explore hybrid systems to address ongoing challenges like evasion techniques.

2 Introduction

Generative AI such as ChatGPT, Gemini, etc., can produce highly realistic content in various forms such as images, texts, and audio. While Gen-AI offers numerous societal benefits, it also raises significant ethical concerns; the proliferation of generative AI models has led to an explosion in AI-generated content, raising concerns about authenticity, misinformation, and plagiarism, enabling individuals to falsely claim copyright ownership of AI-generated content.

To address the aforementioned issues with Gen-AI, AI detection software has been introduced to determine whether some content (text, image, video, or audio) was generated using artificial intelligence. However, this development has hit a setback as many issues have been raised about the precision: to say, the software isn't always reliable [1].

In a 2023 study by Weber-Wulff et al., researchers evaluated 14 detection tools, including Turnitin and GPTzero, and observed that "all scored below 80% of accuracy and only 5 over 70%". They also observed that these tools tend to have a bias for classifying texts more as human than AI, and that the accuracy of these tools worsens upon paraphrasing, i.e., post-processing [2]. Owing to this, two outcomes have been outlined as a result of the unreliability of such tools, namely;

- **False Positives:** In AI content detection, a false positive is when a human-written work is flagged incorrectly as AI-written. False positives in an academic setting frequently lead to accusations of academic misconduct, which can result in serious consequences. Many AI detection platforms claim to have a minimal level of false positives; for example, Turnitin claims less than 1% false positives [3]. However, later research by 'The Washington Post' produced much higher rates of 50%, albeit a smaller size was used in the research [4].
- **False Negatives:** A False Negative is the failure to identify documents with AI-written texts. False Negatives often happen as a result of the deployment of evasive techniques in the generation of the content to make it sound more human [5] or the sensitivity of the detection software being used.

Following the release of ChatGPT and similar AI text generative software, many educational

institutions have issued stern policies against the use of AI by students [6]. For text, this is usually done to prevent plagiarism, often by detecting repetition of words as telltale signs that a text was generated by an AI tool (including hallucinations). As stated earlier, current detectors may sometimes be unreliable and have incorrectly marked works by humans as originating from AI [7], while failing to detect AI-generated work in other instances [8].

Thus, to improve the reliability of AI text detection, researchers have explored digital watermarking and deep-learning transformer techniques. These approaches enable content to be accurately flagged as AI-generated, even when it has been slightly paraphrased or modified.

This study explores the two main current detection methods, categorized as reactive (classifier-based) and proactive (watermarking at generation). It seeks to compare their precision and wholesomeness vis-à-vis available datasets and relevant literature. Key terms: Robustness means how well a method resists changes like paraphrasing, while accuracy is the percentage of correct identifications, including avoiding false positives (human text flagged as AI) and false negatives (AI text missed).

3 Literature Review

3.1 Watermarking Techniques

Watermarking is the process of embedding a visible or invisible identifier [9], such as a logo or a unique code, into digital content to indicate ownership and guard against unauthorized use. It serves as a means of identification and provides a layer of protection for digital assets. It can also be used to verify the authenticity or integrity of the content. Originally, the term 'watermark' came from the paper industry, where a faint mark was pressed into the paper during the manufacturing process. Due to its potential, watermark-based detection has garnered cognitive attention over the years. For instance, an Executive Order issued by the White House in October 2023 recommended watermarking AI-generated content. In the same vein, the EU AI act on July 2024 suggested ethical guidelines for AI trustworthiness. As such, several major companies have already deployed watermark-based detection. For example, OpenAI embeds visible watermarks into images generated by DALL-E, and Microsoft watermarks all AI-generated images created via Bing.

Watermarking is a proactive approach; specifically, a watermark is embedded into all content generated by a Gen-AI service/platform, and a piece of content is identified as AI-generated if a similar watermark can be decoded from it. Cao (2025) reviewed the methods across modalities, probabilistic for text (adjust token distributions), spatial/frequency-domain for images, and spectral/temporal for audio. The challenges outlined included adversarial attacks and standardization [10]. Fu et al. (2024) proposed semantic-aware watermarking for conditional text generation, reducing hallucinations while maintaining detectability (z-scores > 4) [11]. Jiang et al. (2024) in their study introduced user-level attribution via unique watermarks, optimizing selection for performance. This supposedly achieves near-perfect accuracy without post-processing, robust to common edits [12].

However, in Jiang et al. (2023), their study suggested evasion via adversarial post-processing (WEvade), adding small perturbations to remove watermarks, which is effective in white/black-

box settings [13].

3.2 Transformer-Based Detection

Detecting AI-generated content, especially texts, is a critical challenge, and transformer-based techniques have seemingly become the forefront of this field. Fundamental models include BERT, RoBERTa, and DeBERTa architectures for classification. They essentially "fight fire with fire", using the same underlying architecture that generates the text (like GPT) to also detect it. Simply put, Transformer-based detection primarily relies on fine-tuning powerful discriminative models on binary classification tasks and supplemented statistical analysis.

Khan et al. (2025) used DistilBERT, achieving 98% accuracy on 500 thousand essays [14]. Self-detection captured linguistic patterns. Raza et al. (2025) applied BERT with progressive training for fake news (related to AI-generated content detection), achieving 95.3% accuracy on WELFake (72000 articles) [15]. These studies were focused on texts, not images, sounds, and videos.

3.3 Datasets

The primary dataset used is the AI-vs-Human Dataset by Shayan Gerami, available at `https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text`. This dataset contains about 500,000 essays created by humans and AI. A watermarked text was also generated using open-source LLMs (GPT-2) with a watermarking library.

4 Methodology and Technical Implementation

To fairly evaluate both methods within a High-Performance Data Analytics (HPDA) context, a 'Hybrid Strategy' pipeline was implemented. The motive behind this development was that standard datasets (like the Kaggle AI vs. Human Text dataset utilized in this study) consist of AI texts generated without watermarks, and the evaluation was split into two distinct, parallel streams to ensure a valid benchmark for both the active and passive methods.

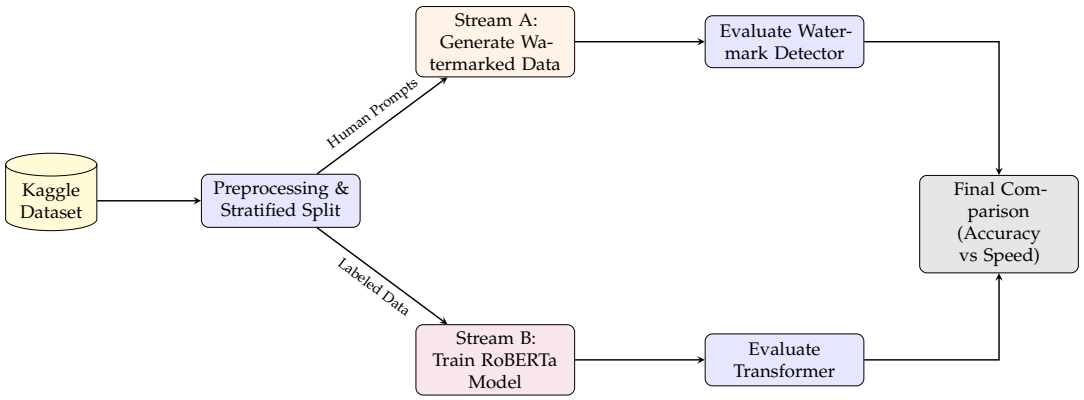

Figure 1: Dual-stream pipeline ensuring valid testing for both active (Watermark) and passive (Transformer) methods.

4.1 Stream A: Watermarking Benchmark Implementation

The watermarking approach focuses on proactive detection. Using a predefined set of human prompts, fresh AI text was generated using a Green-Red list algorithm implemented atop a pre-trained GPT-2 model. The parameters were set to a watermark strength of $\delta = 2.0$, a green-list proportion of $\gamma = 0.5$, and a z-score detection threshold of 4.0.

The core of the detection benchmark relies on hashing the previous token to deterministically seed a random number generator, which then selects the 'green list' for the current token. The text is scanned sequentially to count occurrences of green-list items and calculate a z-score. Below is the core logic used in the implementation to evaluate the watermark's presence:

```
# Tokenize text and count green tokens
tokens = self.tokenizer.encode(text)
green_count = 0

for i in range(1, len(tokens)):
    previous_token = tokens[i-1]
    current_token = tokens[i]

    green_list = self.get_green_list(previous_token, secret_key)
    if current_token in green_list:
        green_count += 1

# Calculate statistical z-score
n = len(tokens) - 1
expected_green = n * self.gamma
std_green = np.sqrt(n * self.gamma * (1 - self.gamma))
z_score = (green_count - expected_green) / std_green
```

Listing 1: Core Z-Score Calculation in Watermark Detector

4.2 Stream B: Transformer Benchmark Implementation

The reactive detection stream utilized a RoBERTa-base model. The model was fine-tuned on a stratified subset of 2,000 samples from the Kaggle dataset. To accommodate HPDA requirements, the HuggingFace Trainer API was utilized with PyTorch, allowing for hardware acceleration and batched gradient descent. The model was trained over 3 epochs with a batch size of 8, enhancing the utilization of mixed-precision where applicable to optimize the memory footprint. Subsequently, both streams were evaluated using a unified metrics pipeline calculating Accuracy, F1-Score, Training Time, and inference performance (measured in samples per second).

5 Evaluation and Discussions

The experimental results revealed a rather fascinating trade-off between the two methods when constrained by computational resources and data scale.

5.1 Experimental Results

As shown in Figure 2, the benchmark yielded distinct profiles for both techniques:

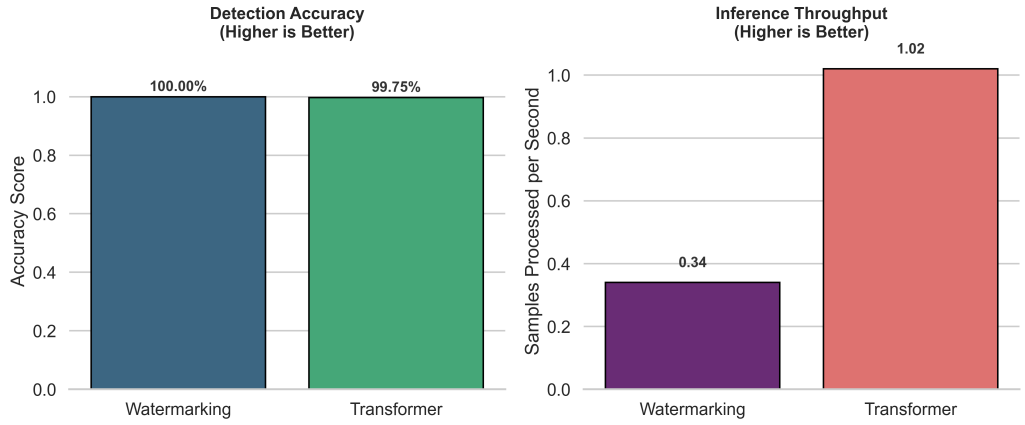

Figure 2: Comparison of Detection Accuracy and Inference Throughput between Watermarking and Transformer methods.

- Watermarking achieved a flawless 100% accuracy and a perfect F1-score of 1.0. Because it relies on mathematical verification rather than deep learning, it required zero training time. However, its inference processing speed was relatively slow at 0.34 samples per second.
- Transformer (RoBERTa) achieved an exceptionally high accuracy of 99.75% with an F1-score of 0.997. Reaching this convergence, however, incurred a massive computational cost, requiring approximately 10.5 hours of training time on a standard CPU. Interestingly, once trained, the model processed data almost three times faster than the watermark method, achieving a throughput of 1.02 samples per second.

5.2 Discussion of Benchmarks

These findings highlight several critical dynamics in AI content detection for HPDA environments:

1. **The Accuracy Convergence vs. The "Training Tax":** The results prove that Transformer models are a highly viable alternative to Watermarking, capable of reaching near-perfect accuracy (99.75%). However, this accuracy comes with a heavy training tax. While watermarking is incredibly agile for immediate, cold-start deployment, deep learning models demand vast computational resources upfront to learn the linguistic patterns of AI.
2. **Inference Throughput & Software Optimization:** Intuitively, a statistical method should be "lighter" than a complex 110-million parameter neural network. However, the benchmark proved the Transformer model was approximately three times faster at inference. Analysis of the code reveals the reason behind this observation: the Watermarking detector relies on sequential, token-by-token verification loops in native Python (as seen in Listing 1), which creates a software bottleneck. Conversely, the Transformer leverages highly optimized matrix operations and vectorization via PyTorch's C++ backend to process text in parallel batches. This highlights a key HPDA principle: algorithmic simplicity does not automatically translate to hardware throughput without low-level optimization.

3. **Fundamental Limitations:** Despite its 100% accuracy, the Watermarking method suffers from what I term as Generator Dependency. It only works if the AI model natively cooperates and embeds the secret key during generation. It is entirely blind to text generated by third-party or rogue models. The Transformer, while computationally expensive to train and susceptible to model drift as new AI generators emerge, maintains the potential to generalize and detect uncooperative AI content.

6 Conclusion and Recommendations

This study demonstrates that the choice between proactive Watermarking and reactive Transformer-based detection is not merely a question of accuracy, as both methods can achieve near-perfection under controlled conditions. Instead, the decision hinges on computational resource allocation and the nature of the deployment environment. Transformers incur a heavy upfront training cost but ultimately provide superior batch-processing throughput due to vectorized hardware acceleration. Watermarking eliminates training costs and offers absolute certainty, but suffers from slower token-by-token inference whilst relying heavily on the cooperation of the AI generator.

Recommendations for HPDA Systems:
For large-scale, real-world data pipelines, a **cascading hybrid approach** is strongly recommended. Systems should utilize Watermarking as a "first pass" for internally hosted or verified AI systems where rapid deployment and zero false-positives are required. For unverified, external content (e.g., scraping the web or processing thousands of student essays from unknown sources), optimized Transformers should be deployed as a fallback to efficiently handle the high batch throughput and catch unmarked AI text.

Future Work:
To build upon this study, future research should explore the robustness of these methods against adversarial attacks. Specifically, benchmarking how well the models maintain their accuracy against advanced paraphrasing tools designed to "wash away" watermarks is critical. Furthermore, rewriting the watermark detector's sequential logic in a lower-level language (e.g., C++) could drastically improve its throughput. Finally, extending these HPDA efficiency comparisons beyond text to multi-modal AI content (such as images and audio) will provide a more comprehensive framework for the future of AI detection.

References

[1] Janelle Shane. Don't use AI detectors for anything important. *Fortune*, July 3, 2023.

[2] Debora Weber-Wulff et al. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1):1–39, 2023.

[3] Annie Chechitelli. AI writing detection update from Turnitin's Chief Product Officer. *Turnitin Blog*, May 23, 2023.

[4] Geoffrey A. Fowler. We tested Turnitin's ChatGPT-detector for teachers. It got some wrong. *The Washington Post*, April 1, 2023.

[5] Cynthia Condit. LibGuides: ChatGPT and Generative AI Legal Research Guide: AI Detection Tools. *University of Arizona Law Library*, Retrieved March 20, 2025.

[6] Alex Hern. AI bot ChatGPT stuns academics with essay-writing skills and usability. *The Guardian*, December 4, 2022.

[7] Geoffrey A. Fowler. Detecting AI may be impossible. That's a big problem for teachers. *The Washington Post*, June 2, 2023.

[8] Josh Taylor. ChatGPT maker OpenAI releases 'not fully reliable' tool to detect AI generated content. *The Guardian*, February 1, 2023.

[9] John Kirchenbauer, Jonas Geiping, et al. A Watermark for Large Language Models. *arXiv preprint arXiv:2301.10226*, 2023.

[10] L. Cao. Watermarking for AI Content Detection: A Review on Text, Visual, and Audio Modalities. *arXiv preprint arXiv:2504.03765*, 2025.

[11] Y. Fu, D. Xiong, and Y. Dong. Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18003–18011, 2024.

[12] Z. Jiang, M. Guo, Y. Hu, and N.Z. Gong. Watermark-based Attribution of AI-Generated Content. *arXiv preprint arXiv:2404.04254*, 2024.

[13] Z. Jiang, J. Zhang, and N.Z. Gong. Evading watermark based detection of ai-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1168–1181, 2023.

[14] H.U. Khan et al. Identifying artificial intelligence-generated content using the DistilBERT transformer and NLP techniques. *Scientific Reports*, 15(1):20366, 2025.

[15] N. Raza et al. Enhancing fake news detection with transformer-based deep learning: A multidisciplinary approach. *PLoS ONE*, 20(9):e0330954, 2025.