2024 International Conference on Emerging Systems and Intelligent Computing (ESIC) | 979-8-3503-4985-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ESIC60604.2024.10481602

Detection of AI-Generated Text Using Large Language Model

Manish Prajapati
School. of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India 751024
Manishraj7719@gmail.com

Santos Kumar Baliarsingh*
School. of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India 751024
santos.baliarsinghfcs@kiit.ac.in

Chinmayee Dora
Department of Electronics & Communication Engineering
Centurion University of Technology and Management
Bhubaneswar, India 752050
chinmayee.dora@gmail.com

Ashutosh Bhoi
Department of Computer Science & Engineering
GITAM Deemed to be University
Visakhapatnam, India 530045
Email: imashubhoi@gmail.com

Jhalak Hota
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India 751024
Email: jhalak.hota@gmail.com

Jasaswi Prasad Mohanty
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India 751024
Email: jasaswi.mohantyfcs@kiit.ac.in

***Abstract*—A large language model (LLM) is a trained deep-learning model that understands and generates text in a human-like fashion. Due to the significant advancements of LLM, it becomes a challenging task to distinguish human-written content from artificial intelligence (AI) generated content. In this work, we leverage the machine learning (ML) models to reliably identify whether an essay is authored by a human being or by an LLM. Concerns about LLMs replacing human tasks, especially in education persist. However, optimism remains for their potential as tools to enhance writing skills. An academic worry is LLMs facilitating plagiarism due to their extensive training in text and code datasets. Using diverse texts and unknown generative models, we replicate typical scenarios to encourage feature learning across models. In a study involving human subjects, we demonstrate that the annotation scheme offered by generative textual likelihood ratio (GLTR) enhances the human detection rate of fake text from 74% to 99% without requiring any previous training. GLTR is open source and publicly deployed, already finding widespread use in detecting generated outputs.**

***Index Terms*—LLM, AI, Machine Learning, ChatGPT, text detection**

I. Introduction

In recent years, AI has made remarkable strides, advancing from generative models in computer vision to LLMs in natural language processing (NLP) [1]. LLMs can now generate high-quality texts with a wide range of applications. For instance, the latest ChatGPT model [2] is capable of producing human-like texts for various activities, including authoring computer program code, composing music lyrics, completing documents, answering questions, and showcasing limitless possibilities. According to the ongoing trend in NLP, these LLMs are expected to improve over time. However, there is a significant challenge concerning authenticity and adherence to rules. The potential for human misuse of AI technologies, such as plagiarism, the generation of false news, spamming, crafting fake product evaluations, and manipulating online content for social engineering, poses a substantial risk to society. We acknowledge that this work will promote open research and transparency regarding AI detection approaches applicable in real-world scenarios. Could assist us in developing a model capable of distinguishing between essays authored by middle and high school students and those generated by LLMs? With the increasing prevalence of LLMs, there is a growing concern that they might replace or alter tasks traditionally carried out by humans. Educators, in particular, express concerns about potential impacts on students' skill development, although some maintain optimism that LLMs could ultimately serve as valuable tools to enhance student's writing proficiency. At the forefront of academic concerns about LLMs is their potential to facilitate plagiarism. LLMs are trained on extensive datasets of text and code, enabling them to generate text very similar to human-written content. For instance, students could utilize LLMs to create essays that are not their own, thereby missing essential learning milestones. Our work in this competition can contribute to identifying distinctive LLM artifacts and advancing the state of the art in LLM text detection. By employing texts of moderate length covering various subjects and generated by multiple unknown models, we aim to replicate typical detection scenarios and encourage the learning of features that generalize across models.

Modern advancements in Natural Language Generation (NLG) methods have amplified the diversity, potency, and value of texts generated by LLMs. OpenAI's ChatGPT stands out as a notable example, demonstrating exceptional performance in tasks ranging from query answering to writing emails, essays, and code. However, this heightened ability to generate human-like text efficiently also brings about concerns related to detecting and preventing LLM abuse in activities like phishing, disinformation, and academic deception. Consequently, many educational institutions have prohibited the use of ChatGPT due to worries about assignment cheating [3], and media outlets have issued warnings about the potential dissemination of misleading news by LLMs [4]. Instances of LLM

misuse have even led to the blocking of NLG implementation in critical sectors, including the field of methodology and education. The ability to accurately identify LLM-generated text is essential for fully realizing the potential of Natural Language Generation (NLG) while mitigating potential adverse consequences. From the perspective of end users, detecting LLM-generated text can instill confidence in NLG techniques and drive broader adoption. In the field of machine learning, such detection systems can aid system engineers and researchers in monitoring generated content and curbing illegitimate use. Given its significance, there is a growing interest among academics and industries to collaborate on the identification of LLM-generated research text, to gain a better understanding of its underlying mechanics. While ongoing debates persist about the adequacy and methods of identifying LLM-generated texts, we provide a comprehensive specialized overview of existing detection approaches. These approaches can be broadly categorized into two groups: pitch-black-box detection and colorless-box detection. Pitch-black-box detection relies on Application Programming Interface (API)-level access to LLMs and involves constructing a classification model using human and machine text samples. This model can effectively distinguish between LLM-generated and human-generated texts, as modern LLM-generated documents often exhibit distinctive language or statistical patterns. However, pitch-black-box approaches may become less successful as LLMs continue to evolve and improve. On the other hand, colorless-box detection provides the detector with complete access to LLMs, allowing regulation of the model's epoch manners for traceability reasons. In practice, pitch-black-box detectors are often developed by external companies, while clear detection is typically performed by LLM.

Our contributions to the current work are summarized as follows:

Our proposed article addresses the problem at hand from the standpoints of machine learning and NLP. In particular, we first describe pitch-black-box detection techniques in terms of the phases of a data analytics vitality process, which include feature selection, data collection, and the creation of category models. Addressing the issue of identifying AI-generated text is crucial to prevent the misuse of technology for unethical purposes such as plagiarism, the generation of false news, and spamming. However, relying on untrustworthy detectors is not the optimal solution, as a detector with a high false positive rate could cause more harm than good in society. Our research highlights the vulnerability of various detectors to simple practical attacks, such as paraphrasing. More importantly, our results indicate that building reliable detectors in real-world scenarios is challenging, language models would need to compromise their performance to maintain consistent detection accuracy. We aim for these findings to initiate an open discussion in the community regarding the ethical and trustworthy use of AI-generated text. Next, we examine more current developments in clear detection techniques, including inference time watermarks and subsequent watermarks. Lastly, we outline the drawbacks and issues with the current detection investigations and provide recommendations for possible directions for further study. Our goal is to maximize the utilization of LLMs by offering foundational ideas, methods, and case studies for identifying texts created by LLMs.

II. Literature Survey

In recent years, many researchers dedicated themselves to the study of the automatic detection of AI-generated text. Those were: "GTLR: Statistical Detection and Visualization of Generated Text" [5], and "Can AI-Generated Text be Reliably Detected?" [1] From these papers, we were able to glean essential information that lent itself to our understanding of the Paper.

In this section, the authors of the paper titled "Can AI-Generated Text be Reliably Detected?" [1] proposed a versatile framework capable of addressing specific scenarios, such as distinctive writing styles, clever prompt design, or text paraphrasing. Additionally, they extended their investigation to include cases where pseudorandom number generators are employed for AI-text generation instead of true randomness, revealing consistent results with a negligible correction term applicable to all polynomial-time computable detectors. Furthermore, the study demonstrates that even large language models (LLMs) protected by watermarking schemes can be vulnerable to spoofing attacks, wherein adversarial humans deduce hidden LLM text signatures and integrate them into human-generated text, making it detectable as text generated by the LLMs. This susceptibility could potentially lead to reputational damage for the developers. It is important to note that the limitations of this approach stem from its testing solely on binary-class datasets. Various methodologies have been developed to detect AI-generated text, including binary classification models, zero-shot classifiers [6], neural network-based detectors like OpenAI's RoBERTa [7], and soft watermarking [8]. Binary classification models are specifically designed to differentiate between AI-generated and human-written text by training on labeled datasets containing examples from both categories. However, their effectiveness is contingent on the quality and representativeness of the training data, and they may encounter challenges with sophisticated AI-generated text that closely mimics human writing. This paper has focused on GLTR [5], a tool to support humans in detecting whether a text was generated by a model. GLTR applies a suite of baseline statistical methods that can detect generation artifacts across common sampling schemes. A human-subjects study shows that the annotation scheme provided by GLTR improves the human detection rate of fake text from 54% to 72% without any prior training. GLTR is open-source and publicly deployed and has already been widely used to detect generated outputs.

Several researchers have utilized established text-matching software to evaluate the presence of plagiarism in AI-generated text. In a study conducted by Khalil and Er (2022) [9], 50 essays generated by ChatGPT were input into two distinct text-matching software systems—25 essays into iThenticate and 25 into the Turnitin system. Notably, both systems serve as different interfaces to the same engine. Their findings revealed

that 40 of the essays (80%) were considered to exhibit a high level of originality, defined by a similarity score of 20% or less. Furthermore, [9] investigated ChatGPT's capability to discern whether the essays were generated by the same AI system. They reported an accuracy rate of 92%, as they classified 46 essays as potential instances of plagiarism. Tests conducted by van Oijen (2023) [10] revealed that the overall accuracy of tools in identifying AI-generated text was only 27.9%, with the most effective tool achieving a maximum accuracy of 50%. In contrast, these tools exhibited a significantly higher accuracy, nearly 83%, when detecting human-written content. The author concluded that the performance of detection tools for AI-generated text was comparable to that of random classifiers [10]. Furthermore, the tests yielded intriguing findings; for instance, the tools encountered difficulty in detecting a piece of human-written text that had been rephrased by ChatGPT or a text passage composed in a specific style. Notably, there were no instances of misattributing human-written text to AI-generated text, indicating an absence of false positives.

III. Proposed Work

A. Dataset

We gathered our data from a publicly available Kaggle website dataset known as "Human-Written Essays (HRE)." The dataset includes essays written by students and others generated by various LLMs. The essays consist of questions accompanied by expert human answers and responses from ChatGPT-3. The group formulated 10,000 questions and recorded corresponding answers. All essays were written in response to one of seven prompts, where students were instructed to read source texts and craft a response. Expert answers were derived from two primary sources: open question-answering datasets and Wikipedia text. The dataset encompasses various question subcategories. Specifically, there were 3,933 questions with engineering essays, including 1,634 human answers and 2,053 AI answers. For medical essays, there were 913 questions, with 1,248 human answers and 1,337 AI answers. Open (QA) questions totaled 1,035, with 1,035 human answers and 2,356 AI answers. Questions from Reddit numbered 5,658, yielding 25,678 human answers and 11,675 AI answers. Additionally, there were 760 questions from Wikipedia, featuring 887 human and 887 AI answers. We chose this dataset so that our step would be concentrated on the experiments rather than the data collection method itself.

B. The Pitch-Black Box Detection

Figure 2 illustrates the limited access to the LLM that other entities have at the Application Programming Interface(API) level in the context of pitch-black box detection. pitch-black-box techniques require the collection of text samples from both machine-generated and human sources to construct an effective detector. The next step is to create a classifier that uses pertinent information to discern between the two groups. We draw attention to the three crucial steps involved in pitch-black-box text detection: feature selection, data collection, and classification model implementation.

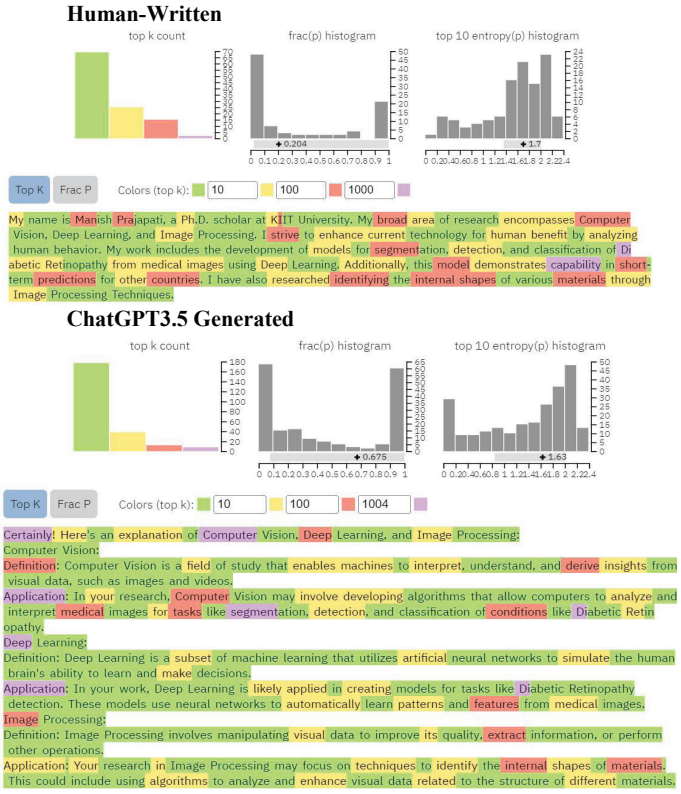

Fig. 1. The word ranking is produced using the Chat-GPT-3.5 small model and is visualized in the result a gltr. The maximum 10 words are represented in green, the maximum 100 in yellow, the top 1,000 in red, and the remainder in purple. There is one major exception. There is a distinction between the two passages. Texts written by humans are compared to ChatGPT-3.5 text LLMs Generated.

1) Data Acquisition: A significant factor in pitch-black-box detection algorithms' efficacy is the caliber and variety of the collected data. Lately, an increasing amount of studies have focused on collecting LLM replies and contrasting them containing materials written by humans that cover a broad spectrum of domains. This section explores the several approaches for gathering information from machine and human sources.

The LLM-generated Data:

Based on the words that came before, LLMs are made to predict the likelihood of tokens that will come after them in a sequence. Improvements in natural language generation have paved the way for the construction of LLMs in several fields, including creating stories, answering questions, and generating news. It is important to establish goal domains, and generation models before acquiring texts generated by LLMs. Generally, a detection model is assembled to identify text generated by a distinct LLM across different fields. Conversely, stochastic methods such as nucleus sampling are better suited for free-form generation, as they maintain a certain level of randomness while excluding insufficient candidates. Finally, while collecting text created by LLMs, academics must carefully assess the target domain, generation models, and sample approaches to ensure the creation of high-quality,

varied, and domain-relevant material.

Human Authored Data: Data that has been manually composed by people is naturally sourced from genuine human authors. For instance, Dugan et al.'s study [11] aimed to assess how well natural language production systems performed as well as how humans perceived the texts that were produced. To do this, they hired 200 people on Amazon Mechanical Turk to annotate ten websites and provide a natural language explanation for their selections. For bigger datasets, manual data collection by human labor might be prohibitively expensive and time-consuming. A different approach is to take content straight out of human-written sources, including academic journals and web pages. As an illustration, we may easily get hundreds of definitions of computer science concepts from chatGPT and other AI tools, which were authored by skilled human professionals. Furthermore, several publicly available benchmark datasets already provide human-authored texts in a structured way. One such dataset is ELI5 [12], which consists of 640K discussions from the Reddit community (explain like I'm five). By making use of these easily accessible resources, gathering texts written by humans may be completed much more quickly and cheaply.

2) Human outcomes of evaluation: Our research has generated significant findings regarding the differentiation of writings created by LLMs from those authored by humans, based on human assessments. According to preliminary findings, papers authored by humans frequently employ syntax and punctuation to express subjective sentiments, whereas those generated by LLMs tend to be less objective and emotive. Human writers commonly use utterance points, query impacts, and ellipses to convey their feelings, while LLMs provide more formal and organized responses. However, it is important to note that texts caused by LLMs might contain false information, making them potentially unreliable or unhelpful.

Studies have demonstrated that human-written papers are typically more cohesive at the sentence level than texts created by LLMs, which often reuse phrases within a paragraph. These findings suggest that LLMs could leave distinct signals in the text they produce, enabling the identification of specific characteristics to distinguish between LLM-generated and human-written.

3) Feature Selection for Detection: In what ways can we differentiate between LLM-generated texts and human-authored texts? This section will explore potential detection features from various perspectives, encompassing statistical imbalance, language pattern, and fact verification.

Statistical Imbalance : The statistical imbalance between LLM-generated and human-authored texts can be detected using different metrics, such as the non-compliance coefficient for gauging conformity to an exponential curve. If we assume that most systems draw samples from the distribution's head, the visualization tool GLTR reveals generation artifacts, allowing for the discernment of LLM-generated text through word ranking information. A widely used criterion called perplexity, which evaluates the uncertainty in predicting the next word, assigns poor ratings to LLM-generated text as it focuses on

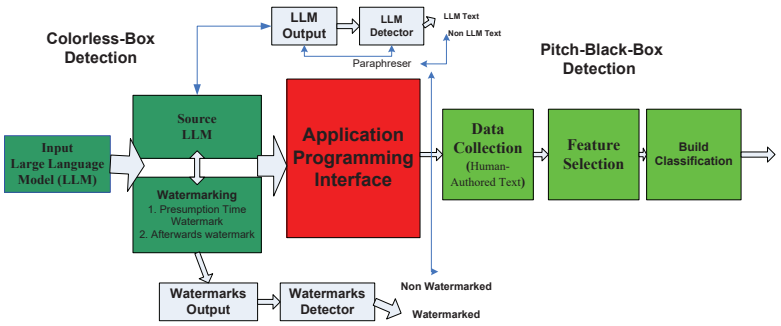

Fig. 2. A basic introduction of LLM-generated text detection .

familiar patterns. However, as illustrated in Figure 1, The need for document-level text constrains these statistical imbalances, resulting in a reduction of detection resolution.

Language Pattern: Language patterns in both human and LLM-generated writings can be analyzed using various contextual aspects, including sentiment evaluation, dependency parsing, part-of-speech, vocabulary features, and stylistic components. Word usage patterns can be inferred from vocabulary parameters such as word density, average word length, and vocabulary size. Previous ChatGPT experiments indicate that texts authored by humans tend to be briefer but employ a more extensive vocabulary. Part-of-speech analysis indicates that nouns are frequently used in ChatGPT messages, suggesting objectivity and argumentativeness. The study on dependency parsing reveals a higher usage of determiners, conjunctions, and auxiliary relations in ChatGPT texts. Sentiment analysis is utilized to gauge the emotional style and philosophy of the text. Unlike humans, LLMs cannot convey emotions and default to neutrality in their communication. Studies show that, in comparison to texts written by humans, ChatGPT communicates much less hate speech and negative emotions.

Repetitiveness, aimlessness, and intelligibility are examples of stylistic elements that serve as useful cues for identifying works produced by LLMs. In addition to examining individual texts, a multitude of language patterns are discovered in multi-turn discussions, providing useful characteristics for identifying material produced by LLMs and reflecting their training set and tactics. It is important to recognize, nevertheless, that LLMs are capable of making significant language pattern adjustments when prompted. For example, the tone and manner of the LLM's response might change when a cue such as "Please respond with humor" is included, affecting the robustness of the language patterns.

Fact Verification: LLMs usually cause nonsensical or inconsistent text due to likelihood maximization objectives in training, known as hallucination. Fact verification becomes crucial for detection, as seen in cases like ChatGPT producing false scientific abstracts. Popular decoding methods like top-k and midpoint selection result in miscellaneous but less verifiable epochs. Fact verification tools and algorithms have been developed, using sentence-level evidence, graph structures, and

knowledge graphs. Assessing consistency and identifying non-factual information, these methods offer the potential to detect LLM-generated texts alongside other features to distinguish them from potentially misleading human-authored texts.

C. Classification Models

Typically approached as a binary classification problem, the detection task seeks to identify textual features that differentiate between human-authored and LLM-generated texts; this section, in turn, aims to provide a comprehensive outline encompassing the primary varieties of category models employed for this purpose.

1) Classification Algorithms: Classification algorithms use a variety of characteristics listed in Section B(3) to differentiate documents created by LLM from those written by humans. Decision trees, Support Vector Machines (SVM), Naive Bayes, and K-nearest neighbors (K-nn) are examples of supervised learning algorithms that are often used. For instance, using statistical and language features, linear regression, SVM, and random forest models were used to correctly recognize texts produced by GPT-2, ChatGPT-3.5, GPT-3, and Grover models. Similarly, using a mix of bigram and Term Frequency - Frequency-inverse Document Frequency (TF-IDF) co-occurrence features in a logistic regression model to show strong performance in detecting GPT-2-generated texts. Moreover, research has shown that applying SVM for classification after pre-trained language models to yank semantic textual characteristics can outperform relying just on statistical data. These algorithms have the benefit of being interpretable, which allows students to examine the significance of input variables and comprehend the reasoning behind the model's categorization of texts as LLM-generated.

2) Deep Learning Approaches: Although deep learning techniques frequently produce better detection results, interpretability is severely limited by their pitch-black-box design. As a result, to understand the reasoning behind the model's conclusions, researchers usually depend on interpretation tools.

D. The colourless-box Detection

An employing colorless box detection, the detector acquires complete access to the target language model, simplifying the integration of hidden watermarks into its outputs and facilitating monitoring for any illicit or suspicious activity. We first describe the three requirements for watermarks in NLG in this section. Next, we give an overview of the two main categories of colorless-box watermarking techniques: after the fact both inference time watermarking and watermarking.

1) Watermarking: We outlined three essential conditions for NLG watermarking, building on earlier studies in traditional digital watermarking: (1) Effectiveness: The watermark must be verifiable and successfully integrated into the created texts while maintaining the standard of the generated texts. (2) Secrecy: The watermark should be intended to be stealthy without causing noticeable changes that might be easily found by classifiers that are automated. It should be impossible to tell it apart from texts that aren't watermarked. (3) Sturdiness: The watermark should be durable and challenging to erase.by typical adjustments like the substitution of synonyms. To remove the watermark, adversaries would have to make substantial changes that would make the texts useless. These three conditions serve as the cornerstone for Natural language generation watermarking and ensure that texts produced by LLM may be traced back.

2) Afterwards Watermarking: A concealed message or identification will be embedded into an LLM-generated text by use of Afterwards watermarks. The concealed message inside the dubious text can be retrieved to validate the watermark. Afterwards watermarking approach may be divided into two basic categories: governed by rules and methods based on neural networks.

Rule-based Approaches: In the beginning, researchers studying natural language used methods from multimedia watermarking, which mostly depended on character changes and were non-language.

Neural-based Approaches: Neural-based techniques perceive the hiding information process as an end-to-end learning procedure, contrasting with rule-based alternatives that necessitate significant engineering effort for design. Typically, these methods involve three elements: an embedding

three networks: an encoder network, a discriminator network, and a watermark decoder network [13]. Presented with a secret and an aim text message (for instance, arbitrary binary digits), the watermark encoder network creates an altered text with the secret message included.

IV. RESULTS AND DISCUSSION

In our research, we frequently assess detection performance using measures such as accuracy or AUC. However, these metrics prove insufficient for security analysis as they only consider typical cases. Let's compare two detectors: Detector A identifies a decisive 75% of texts produced by LLM but exhibits a 90% probability of success on the remaining ones at random. On the other hand, Detector B achieves a 99.87% success rate on all data. While both detectors possess identical AUC or detection accuracy in Table 1, the ROC AUC training measures in Table III, reveal that Detector B is relatively weak, whereas Detector A demonstrates remarkable potency.

It is crucial for detectors to accurately recognize text created by LLM, considering a low false-positive rate (FPR) system. Striving for a low false-positive rate and a high True-Positive Rate (TPR) is essential in computer security. The goal of developing techniques around minimal false-positive regimes is frequently emphasized, particularly for groups like non-native speakers who generate non-standard material. Such groups may be more vulnerable to false-positive results, which could have detrimental effects if these detectors are utilized within our academic institutions. The ROC AUC testing measures in Table IV, further demonstrate the performance of these detectors. Table II showcases key public benchmarking datasets for detecting LLMs across diverse domains. Despite

numerous studies on LLM-generated text detection, a comprehensive benchmarking dataset for performance comparison is lacking due to variations in detection targets. Rapid LLM evolution adds to the challenge, with new models emerging monthly, making it difficult for researchers to keep up and create datasets reflecting each model accurately. The ongoing challenge is to establish adaptable benchmarking datasets to accommodate the swift influx of new LLMs.

TABLE I
CONFUSION MATRIX

Sample	precision	recall	f1-score
Human-Written (0)	0.98	1.00	0.99
LLM-Generated (1)	1.00	0.99	0.99
accuracy			99.89

V. CONCLUSION

In this paper, we implemented the regression machine learning model on a classification problem to detect human and machine-generated text. We compared it to earlier models with a unidirectional architecture, highlighting the superiority of its bidirectional design, allowing tokens to grasp context from both preceding and succeeding tokens. The field of text detection created by LLMs is continuously expanding and evolving, with numerous novel methods emerging regularly. This study provides a detailed analysis and accurate classification of the current methods. Without being affected by the rapid progress in LLM-generated text detection, there remain several crucial issues to address. The performance measures for ROC AUC testing are 99.89%, and without being affected by performance measures for ROC AUC training are also very high at 99.82%. To advance in this field, it is imperative to devise innovative solutions to overcome these challenges.

REFERENCES

[1] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can ai-generated text be reliably detected?" *arXiv preprint arXiv:2303.11156*, 2023.

[2] T. OpenAI, "Chatgpt: Optimizing language models for dialogue. openai," 2022.

[3] R. Tang, Y.-N. Chuang, and X. Hu, "The science of detecting llm-generated texts," *arXiv preprint arXiv:2303.07205*, 2023.

[4] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

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

[6] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "Detectgpt: Zero-shot machine-generated text detection using probability curvature," *arXiv preprint arXiv:2301.11305*, 2023.

TABLE II
COMPARISON WITH PAST RESEARCH

S.No	Datasets	LLMs	Roc Auc
1	TURINGBENCH [14]	GPT1,2,3	97.76%
2	TweepFake [15]	GPT2	97.97%
3	Neural Fake News [16]	Grover	98.29%
4	GPT2-Output [17]	GPT2	98.34%
5	HC3 [18]	ChatGPT	98.99%
6	**ESSAYS**	**ChartGPT**	**99.89%**

TABLE III
THE PERFORMANCE MEASURES ROC AUC TRAINING

Step	Training Loss	Validation Loss	Roc Auc(%)
230	No log	0.238509	0.993966
460	No log	0.391851	0.995678
690	0.140200	0.308120	0.995183
920	0.140200	0.224252	0.998052
1150	0.050800	0.295393	0.997867
1380	0.050800	0.238047	0.998277

TABLE IV
THE PERFORMANCE MEASURES ROC AUC TESTING

source	Roc Auc (%)
llama_70b_v1	0.9969105289174494
mistral7binstruct_v1	0.9989151398264224
mistral7binstruct_v2	0.9937575030012005
chat_gpt_moth	0.9989151398264224
radek_500	0.9949095022624435
falcon_180b_v1	0.997526113249038
darragh_claude_v7	0.9974048442906575
darragh_claude_v6	0.9974048442906575
llama2_chat	0.9989195678271309

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

[8] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," *arXiv preprint arXiv:2301.10226*, 2023.

[9] M. Khalil and E. Er, "Will chatgpt get you caught? rethinking of plagiarism detection," *arXiv preprint arXiv:2302.04335*, 2023.

[10] D. Weber-Wulff, A. Anohina-Naumeca, S. Bjelobaba, T. Foltỳnek, J. Guerrero-Dib, O. Popoola, P. Šigut, and L. Waddington, "Testing of detection tools for ai-generated text," *arXiv preprint arXiv:2306.15666*, 2023.

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

[12] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "Eli5: Long form question answering," *arXiv preprint arXiv:1907.09190*, 2019.

[13] S. Abdelnabi and M. Fritz, "Adversarial watermarking transformer: Towards tracing text provenance with data hiding," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 121–140.

[14] A. Uchendu, Z. Ma, T. Le, R. Zhang, and D. Lee, "Turingbench: A benchmark environment for turing test in the age of neural text generation," *arXiv preprint arXiv:2109.13296*, 2021.

[15] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "Tweepfake: About detecting deepfake tweets," *Plos one*, vol. 16, no. 5, p. e0251415, 2021.

[16] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," *Advances in neural information processing systems*, vol. 32, 2019.

[17] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, and L. S. Chao, "A survey on llm-gernerated text detection: Necessity, methods, and future directions," *arXiv preprint arXiv:2310.14724*, 2023.

[18] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, "How close is chatgpt to human experts? comparison corpus, evaluation, and detection," *arXiv preprint arXiv:2301.07597*, 2023.