UNIVERSITY OF BRADFORD

Faculty of Engineering and Digital Technologies

School of Computer Science, AI and Electronics

MSc Applied Computer Science & Artificial Intelligence

COURSEWORK

Module: COS7045-B Advanced Machine Learning

AI-generated text identification using classical machine learning and transformer-based models

Prepared by :

Fouzia Shile (UB: 24008465)

April 22, 2025

Declaration

I attest that this submission is entirely my own work, and all external sources have been properly cited. I fully agree to a plagiarism check. I also wish to be transparent about my use of AI tools to refine the writing style.

Fouzia Shile

Abstract

The growing realistic nature of texts generated by AI using large language models (LLMs) poses significant challenges for authorship authentication. This study introduces a comparative detection framework grounded in the KDD process, evaluating both classical machine learning and transformer-based models on an augmented, diverse dataset. Seven classical models were trained, while BERT, DistilBERT, and RoBERTa were evaluated under fine-tuned and frozen configurations. Results show that classical models, particularly logistic regression, SVM, and MLP outperformed transformer models, achieving F1-scores of up to 0.99 and demonstrating strong generalisation across short, medium, and long essays. In contrast, fine-tuned transformers exhibited instability and class collapse, while frozen RoBERTa achieved a maximum F1-score of 0.97. This work highlights that a classical machine learning model, when trained on well-preprocessed data, can achieve performance comparable to large-scale pre-trained transformers, while offering greater robustness and interpretability in practical detection scenarios.

Keywords: AI-generated text detection, Large language models (LLMs), Machine learning (ML), Natural language processing (NLP), Transformers

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem description

The extensive utilisation of LLMs such as ChatGPT and GPT-4 has revolutionised content creation by enabling the generation of coherent and contextually pertinent material with minimal human intervention. While these advancements offer significant benefits in education, communication, and media, they also raise substantial concerns over the ability to distinguish between human-authored and machine-generated content. This issue is particularly pressing in academic, journalistic, and legal domains, where the validity and traceability of material are paramount. The variable efficacy of current detecting techniques constitutes a significant issue. While numerous systems exhibit satisfactory performance when analysing text from earlier models such as GPT-2 or GPT-3.5, their efficacy diminishes markedly when confronted with the more intricate outputs of later models like GPT-4 (Elkhatat et al., 2023). This may lead to false positives, incorrectly categorising human-generated content as AI-produced, and false negatives, allowing synthetic text to be mistaken for authentic material. Both outcomes undermine the reliability of AI detection systems (Anderson et al., 2023) and lead to unjust consequences. Furthermore, contemporary detectors often exhibit deficiencies in cross-domain, writing style, and language generalisation. These systems can be readily deceived by rudimentary adversarial techniques such as slight alterations in prompts or paraphrase. This vulnerability reveals a significant technological deficiency in current detection methodologies (Yadagiri and Pakray, 2025). Numerous training datasets are constrained in scope, missing diversity and failing to encompass the whole spectrum of natural human writing, hence impacting fairness and model generalisation(Wang et al., n.d.). Recent initiatives have examined deep learning techniques, particularly transformer-based architectures such as BERT, RoBERTa, and BiLSTM, in response to this advancement, as they have demonstrated promising results in modelling linguistic nuances and improving detection accuracy (Wang et al., n.d.; Yadagiri and Pakray, 2025). A significant obstacle in enhancing detection systems is the absence of standardised evaluation criteria across many domains, languages, and adversarial environments (Wu et al., 2025). Overall, identifying language generated by artificial intelligence remains a challenging and rapidly evolving problem. Addressing

it necessitates robust, equitable, and adaptable solutions capable of managing diverse materials, combating manipulation, and maintaining high precision in actual applications.

1.2 Motivation and Objectives

The rise of LLMs has made it increasingly difficult to tell apart AI-generated text from content written by humans. This growing challenge has sparked concern in academic, professional, and policy circles, especially as issues like plagiarism, misinformation, and authorship integrity become more complex. Despite the fact that several detection tools have been developed in response to this situation, many of them still face problems of robustness, domain generalisation and fairness, particularly when applied to sophisticated generative models. This project is motivated by the need for a systematic evaluation of detection models. Specifically, it explores a dual pipeline combining classical ML models and transformer-based models, tested on an extended dataset containing over 46000 samples. The study investigates the performance of models like LR, SVM, and RoBERTa under different feature extraction strategies, including TF-IDF and transformer tokenization. It also analyses robustness across varying essay lengths and the effects of fine-tuning on transformer models for this task. To achieve this, we adhere to the structure presented in Section 1.4.

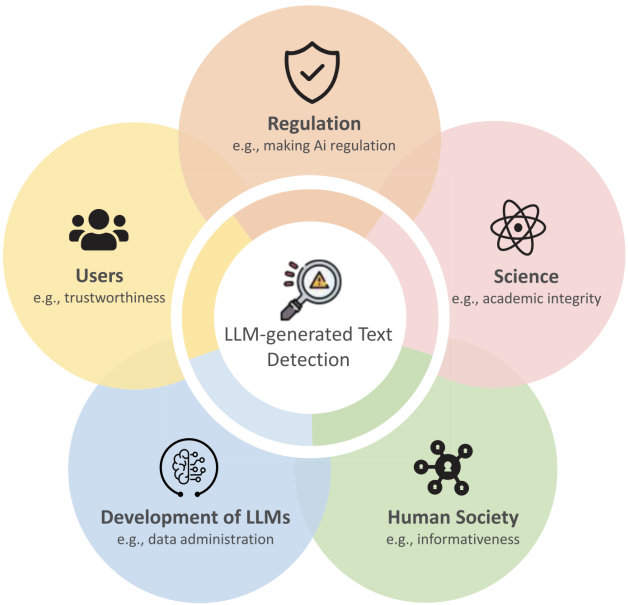


Figure 1.1: Why we need LLM-generated text detection? (Wu et al., 2025)

1.3 Research challenges

Concerning research challenges related to the detection of AI-generated text, and based on the findings reported in (Elkhatat et al., 2023) and (Wu et al., 2025), we

summarize the main challenges encountered in Table 1.1.

1.4 Report Structure

This report is organised into six chapters. Chapter 2 reviews recent research on AI-generated text detection, comparing traditional and deep learning methods while identifying key challenges and research gaps. Chapter 3 outlines the methodology adopted in this study, including dataset selection, preprocessing, feature extraction, and model design. Chapter 4 presents the practical implementation and results of various detection models, supported by evaluation metrics and visual analyses. Chapter 5 offers a critical discussion of the findings, addressing their implications alongside legal, ethical, and social considerations. Finally, Chapter 6 concludes the report by summarising the main contributions and proposing directions for future work.

Table 1.1: Key challenges in detecting AI generated text

Challenge	Description
Accuracy variability	Detection tools exhibit varying performance, with better results on older models (e.g., GPT-3.5) and lower accuracy on newer ones (e.g., GPT-4), leading to inconsistent detection outcomes (Elkhatat et al., 2023).
Increasing model sophistication	As language models advance, their output becomes more human-like, complicating efforts to distinguish AI-generated text from authentic human writing (Elkhatat et al., 2023).
False positives	Human-written content may be mistakenly flagged as AI-generated, which can lead to unfair accusations of plagiarism or misconduct (Elkhatat et al., 2023).
False negatives	AI-generated content can be incorrectly classified as human-written, allowing deceptive or plagiarized material to go undetected (Elkhatat et al., 2023).
Inconsistent tool performance	Different detection tools often provide conflicting results for the same text, undermining trust in their reliability (Elkhatat et al., 2023).
Lack of context awareness	Many detectors fail to account for the context in which the text was created, affecting classification accuracy (Elkhatat et al., 2023).
Rapid evolution of AI	Continuous advancements in LLMs require detection tools to evolve quickly in order to remain effective (Elkhatat et al., 2023).
Dependence on training data	Detection models depend heavily on the diversity and quality of their training datasets, which may not cover all human writing styles (Elkhatat et al., 2023).
Indistinguishability	LLM-generated text often mimics human language convincingly, especially when it includes fabricated details or neutral tones, making detection more difficult (Ji et al., 2023).
Out-of-distribution problems	Detection tools often perform poorly when analysing text from unfamiliar domains, languages, or unseen LLM architectures (Wu et al., 2025).
Potential attacks	Techniques like paraphrasing, adversarial perturbations, and prompt-based manipulation can be used to evade detection systems. (Shi et al., 2023; He et al., 2023)
Real-world data issues	Mixed or human-edited AI content complicates detection, and ambiguous or noisy datasets may limit the effectiveness of detection research (Wu et al., 2025).
Model size impact	Larger language models tend to generate higher-quality, harder-to-detect text, while the size and capacity of detection models influence their generalization and accuracy Wu et al. (2025).
Lack of effective evaluation frameworks	Existing benchmarks are limited and often fail to reflect real-world diversity in models, tasks, languages, and adversarial threats (Wu et al., 2025).

Chapter 2

Literature review

While Chapter 1 introduced the motivation and challenges of detecting LLM-generated text, this chapter reviews the task in technical terms and summarises recent findings from the literature.

2.1 Overview of the AI-generated text detection task

Detecting text generated by LLMs has become a critical research area, driven by the increasing fluency and realism of outputs from models like GPT-4, Claude, and PaLM. The task is typically framed as a binary classification problem,

$$D(x) = \begin{cases} 1, & \text{if } x \text{ is generated by an LLM} \\ 0, & \text{if } x \text{ is human-authored} \end{cases}$$

where the aim is to distinguish AI-generated content from human-authored text (Wu et al., 2025). Studies reveal that humans often perform only slightly better than chance at this task, due to the syntactic fluency and logical consistency of LLM-generated text (Clark et al., n.d.). Linguistic markers such as stylistic simplicity, factual hallucinations, or excessive formality are subtle and inconsistent (Ji et al., 2023; Wu et al., 2025). Detection systems also face practical limitations. Adversarial attacks such as paraphrasing or rewording can degrade model accuracy (Gehman et al., 2020), while performance often drops in cross-domain or unseen-model settings (Feurer et al., 2022). In addition, standardised benchmarks remain limited, making consistent evaluation difficult (Wu et al., 2025). Given the increasing use of LLMs in education, journalism, and content creation, robust detection systems are vital for addressing concerns related to plagiarism, misinformation, and authorship integrity (Wu et al., 2025). This underscores the need for detection approaches that are not only accurate but also generalisable and transparent.

2.2 Prior work on detecting AI-generated text

Various methods have been proposed to detect AI-generated text, including watermarking, statistical, neural, and human-assisted approaches. Table 2.1 summarises

their key characteristics. This project will specifically focus on the exploration and evaluation of ML-based detection methods.

2.3 ML models used in literature for AI generated text detection

Table 2.2 summarises the main ML methods, datasets, and best-performing models reported across selected studies for detecting AI-generated text.

Table 2.2: Summary of ML models applied for AI-generated text detection

Paper	ML Methods Used	Dataset Used	Best Model
(Yadagiri and Pakray, 2025)	Logistic Regression, SVM, Random Forest, CNN-LSTM, RNN, BI-GRU, Fine-tuned BERT, DistilBERT, RoBERTa	Kaggle LLM-Detect-AI Dataset (10k essays, 2 prompts)	Fine-tuned RoBERTa-OpenAI (Accuracy = 0.98)
(Wang et al., n.d.)	Fine-tuned BERT	Private dataset (1378 texts)	Fine-tuned BERT (Training Acc = 98.1%, Test Acc = 97.7%)
(Mindner et al., 2023)	XGBoost, Random Forest, MLP Neural Network	Human-AI Text Corpus (500 samples, 10 topics)	MLP and Random Forest (F1 $\approx$ 97%)
(Hamed and Wu, 2023)	xFakeSci, Naive Bayes, SVM, Logistic Regression	Biomedical abstracts (PubMed + ChatGPT)	xFakeSci (F1 = 91–94%)
(Shah et al., n.d.)	Logistic Regression, Decision Tree, Random Forest, SVM, Gradient Boosting	Custom Wikipedia Dataset (20k texts)	Logistic Regression (92% Accuracy)
(Dou et al., n.d.)	Fine-tuned RoBERTa (Error Span Classification)	SCARECROW Dataset (13k annotations)	Fine-tuned RoBERTa-Large
(Mitchell et al., 2023)	DetectGPT (Zero-shot perturbation detection)	GPT outputs + Human datasets (XSum, SQuAD, WritingPrompts)	DetectGPT (AUROC 0.92–0.99)

2.4 Strengths and limitations of existing approaches

Table 2.3 highlights the key conclusions, strengths, and limitations identified in the reviewed studies.

Table 2.1: Summary of methods for AI-generated text detection

Method Type	Example models	Advantages	Limitations
Watermarking techniques	Statistical watermarking, backdoor watermarking (Gu et al., 2022)	Lightweight integration into model outputs, no retraining needed	Vulnerable to paraphrasing attacks, weaker on short texts
Statistical methods	high-order n-grams, log-likelihood, statistical similarity of bigram counts (Gallé et al., 2021; Solaiman et al., 2019; Hamed and Wu, 2023)	Simple and fast evaluation, interpretable metrics	Poor robustness to domain shift, easily fooled by rewording
Neural-based detection	Linguistic feature-based classifiers, Fine-tuned BERT and RoBERTa, GPTDetector, DetectGPT (Shah et al., n.d.; Dou et al., n.d.; Mitchell et al., 2023)	Captures deeper semantic and stylistic cues, better generalisation	Requires large datasets, computationally expensive
Zero-shot black-box methods	GECScore, AuthentiGPT, Fast-DetectGPT (Wu, Zhan, Wong, Yang, Liu, Chao and Zhang, 2024; Guo and Yu, 2023; Bao et al., 2023)	No need for model internals, transferable across LLMs	Lower confidence, adversarial attacks reduce effectiveness
Instructional Contextual Learning (ICL)	OUTFOX reciprocal training (Koike et al., 2023)	Robust against adversarial attacks, adaptive learning	Computationally intensive, complex setup
Human-assisted detection	Logical anomaly detection, visual cues (Uchendu et al., 2023; Dugan et al., 2023)	Increases detection interpretability, identifies context-based issues	Subjective, inconsistent across different annotators
Feature-based hybrid models	Syntax, punctuation, structural features and XGBoost (Mindner et al., 2023)	Fast training, low resource demand	Brittle for novel LLM generations, weak deep feature extraction
White-box logits-based detection	Log-likelihood and token rank analysis (Solaiman et al., 2019)	Highly accurate when logits are accessible	Requires internal model access (rare for proprietary LLMs)
Prompt rewriting detection	Regeneration similarity testing (Yu et al., 2023)	Can detect manipulation and paraphrase attacks	Dependent on quality of regenerated samples

Table 2.3: Summary of Key Conclusions, Strengths, and Limitations of Reviewed Studies

Paper	Key conclusions	Strengths	Limitations
(Yadagiri and Pakray, 2025)	DL models outperform classical ML.	Comprehensive model comparison, detailed preprocessing.	Dataset limited to two prompts, no adversarial testing.
(Wang et al., n.d.)	Fine-tuned BERT generalizes well on AI-human classification.	Strong generalization, solid preprocessing.	Private dataset, no classical ML baselines.
(Mindner et al., 2023)	Combining handcrafted and novel features boosts detection.	New corpus creation, innovative feature engineering.	English-only, topic-specific, no domain generalization.
(Hamed and Wu, 2023)	xFakeSci outperforms classical ML in biomedical detection.	Novel graph-based features, multi-disease coverage.	Biomedical domain only, specific feature assumptions.
(Shah et al., n.d.)	Classical ML with stylistic features achieves high accuracy.	Focus on explainability (LIME, SHAP), feature-based insights.	Limited to Wikipedia, no deep learning models tested.
(Dou et al., n.d.)	Scaling LLM size improves error type detection.	Fine-grained span-level annotations, open dataset.	News domain only, focus on partial error detection.
(Mitchell et al., 2023)	Zero-shot detection generalizes well without retraining.	No training needed, strong domain transferability.	Needs model log probabilities, computationally expensive.

To sum up, even if recent machine learning techniques have greatly improved the identification of AI-generated text, major issues still exist, especially with respect to domain robustness and dataset diversity. The next chapter describes the approach taken in this project to tackle these issues by means of the design, execution, and assessment of chosen machine learning models.

Chapter 3

Methodology

The general approach taken in this work is organised around the Knowledge Discovery in Databases (KDD) process, which comprises the sequential phases of data selection, preprocessing, transformation, data mining, and evaluation. Figure 3.1 shows a flowchart summarising the main steps taken. considering the study's focus on text classification, separate experimental pipelines were created for classical ML models and DL models, each following tailored preprocessing.

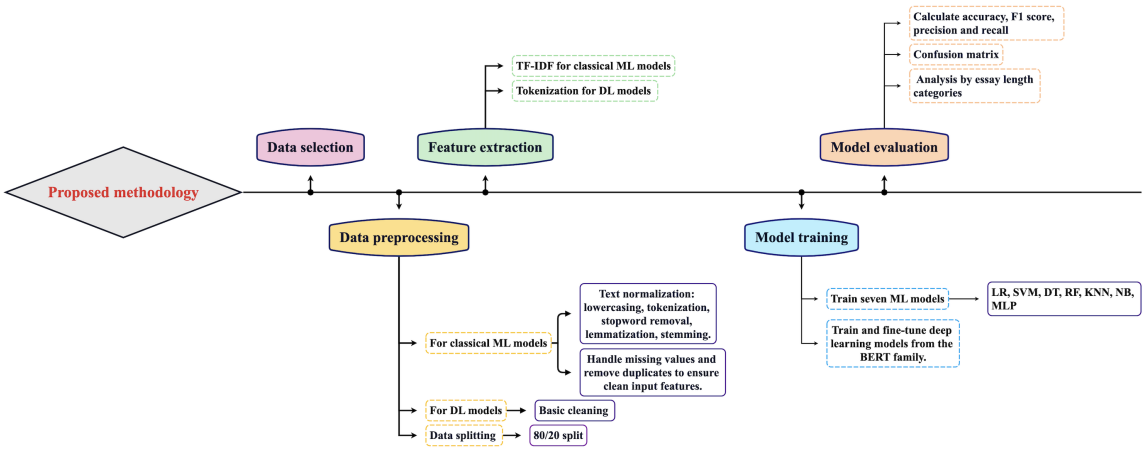


Figure 3.1: Overview of the proposed methodology for AI-generated text detection

3.1 Data description and exploration

This section provides an overview of the dataset used in this study, including its structure, origin, and salient features. This project's is based on the **LLM - Detect AI-Generated Text** dataset, which includes essays written by humans and by LLMs. This dataset is openly accessible on Kaggle. Every essay has a binary indicator attached to it:

- `0`: A human-written essay.
- `1`: AI-generated essay.

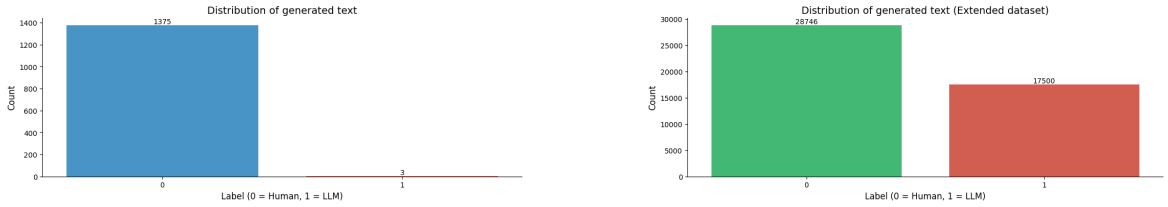
Essays written in response to two different prompts—*"Car-free cities"* and *"Does the electoral college work?"*—are included in the dataset. After aligning column names, two sources were concatenated to create the final training set:

- **dforiginal.csv**: The original dataset from the competition, which included 1378 labelled essays.
- **detectAItext.csv**: An expanded dataset that adds more labelled samples to the original.

The extended dataset (**detectAItext.csv**) introduced many important improvements:

- Inclusion of essays generated by new models such as GPT-4, Google PaLM, and Cohere Command.
- Addition of new writing prompts, with mappings to the original source texts from the persuade corpus.
- Application of quality filtering techniques ("RDizzl3_seven" filtering [1]) to improve data reliability.

After standardizing the column names, both datasets were merged to form a final training set. This operation increased the number of available samples to 46246, while also improving the class balance between human-written and AI-generated essays. Figures 3.2(a) and (b) illustrate the distribution of classes before and after the dataset extension.



(a) Distribution before extension

(b) Distribution after extension

Figure 3.2: Comparison of class distributions before and after dataset extension.

The fundamental characteristics of the dataset are described according to the 4 Vs framework in Figure 3.3:

[1] The RDizzl3_seven filtering technique was designed to remove trivially detectable AI-generated samples by analysing linguistic features such as sentence structure, word diversity, and formatting patterns. Its application results in a more challenging and realistic dataset, fostering better generalization and robustness in text detection models.

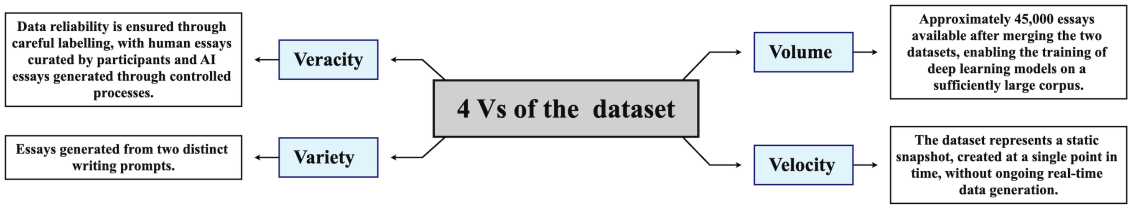
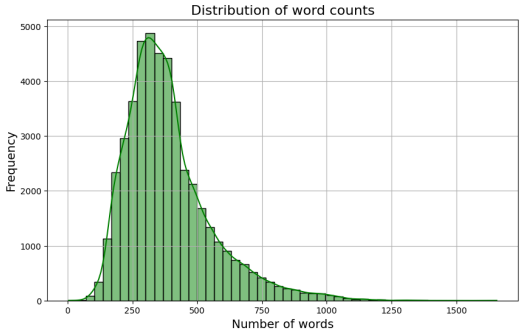
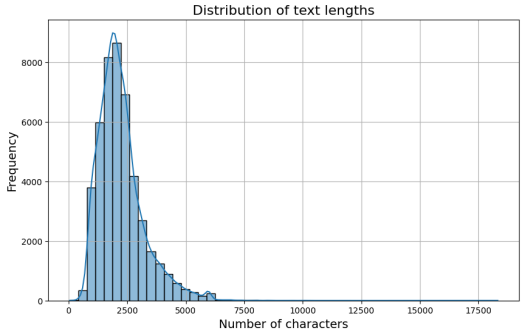


Figure 3.3: 4 Vs characteristics of the final extended dataset.

This final dataset offers a robust foundation for building, training, and evaluating ML and DL classifiers for detecting AI-generated text. Recent research shows that training detectors on diverse and augmented datasets significantly improves robustness against adversarial attacks and model generalization failures (He et al., 2023; Wu et al., 2025). In line with these findings, this project utilized an extended dataset incorporating essays from newer LLMs (e.g., GPT-4, PaLM, Cohere Command) and varied prompts. This natural diversification acts as an implicit augmentation strategy, strengthening the model's ability to detect AI-generated text beyond the original competition distribution.



((a)) Distribution of essay word counts.



((b)) Distribution of essay character lengths.

Figure 3.4: Word and character count distributions for essays in the extended dataset.

The distribution of word counts over essays in the last dataset is shown in Figure 3.4(a). Ensuring a fair variation in essay length that supports strong model training, most essays have between 200 and 500 words. The distribution of character lengths for essays is shown in Figure 3.4(b). Reflecting typical essay structures, most of the essays fall between 2000 and 5000 characters, which is consistent with the distribution of word count.

3.2 Data preprocessing

In this study, separate preprocessing techniques were developed for DL models and classical ML models. While DL models run directly on tokenised raw text representations, classical models needed significant text normalisation and feature engineering. The preprocessing techniques used on each modelling approach are covered in Sections 3.2.1 and 3.2.2.

3.2.1 Preprocessing for classical machine learning models

This section focusses on the KDD framework's preprocessing phase, outlining the procedures used to clean and get the text data ready for traditional ML models listed in Section 3.3.1. Figure 3.5 summarises the whole preprocessing process used for classical ML models. While the earlier phases concentrated on cleaning and standardising the textual data, particular attention was given to the feature extraction step. Specifically, Term Frequency-Inverse Document Frequency (TF-IDF) vectorisation was used to turn the cleaned text into numerical feature vectors. Preliminary tests comparing vocabularies of 5000 and 10000 words were done to optimise feature representation as shown in Table 3.1. Without causing overfitting or notable computational cost, the findings, summarized in Table 3.1, showed modest but consistent increases in classification performance using 10000 terms. Therefore, for all the following experiments, the feature space was defined as the most informative terms 10000.

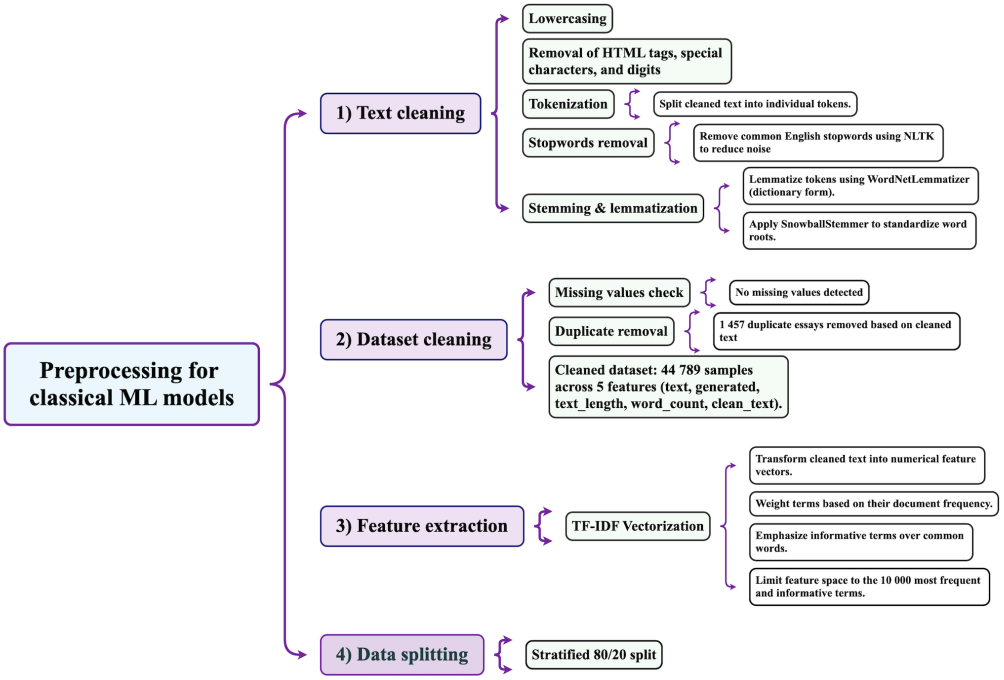


Figure 3.5: Preprocessing pipeline for classical ML models

Class balancing strategy

Given the moderate class imbalance observed (approximately 61% human-written vs. 39% LLM-generated essays, see Figure 3.2), explicit resampling techniques such as oversampling or undersampling were avoided. Instead, class weighting strategies were employed within the models to compensate for the skewed label distribution. This approach preserves the original data distribution and avoids the potential drawbacks associated with resampling methods, such as overfitting due to duplicated samples or information loss from majority class undersampling. Prior research

supports this strategy and oversampling can improve performance in cases of extreme imbalance, (Buda et al., 2018) highlight that class weighting often provides a competitive alternative without introducing synthetic artifacts, especially when the imbalance is moderate. Consequently, class weighting was selected to ensure model robustness while maintaining dataset integrity.

Table 3.1: Comparison of classification performance between 5000 and 10000 TF-IDF features.

Model	Accuracy (5k)	Accuracy (10k)	F1-score (5k)	F1-score (10k)
LR	0.99	0.99	0.99	0.99
SVM	0.99	0.99	0.99	0.99
DT	0.92	0.93	0.90	0.91
RF	0.98	0.98	0.98	0.98
KNN	0.98	0.98	0.98	0.98
NB	0.96	0.96	0.94	0.95
MLP	0.99	0.99	0.99	0.99

3.2.2 Preprocessing for transformer models

In particular for transformer-based architectures like BERT and DistilBERT, this part describes the preprocessing pipeline suggested for deep learning models. Unlike traditional ML models that demand clear feature engineering and text simplification, transformer models expect raw natural language input. The preprocessing thus concentrated mostly on data cleaning and little transformation to preserve the whole contextual richness of the text. The main preprocessing steps are summarised in Figure 3.6. Beginning with simple text normalization and dataset cleaning, the pipeline then integrated directly with pre-trained KerasNLP processors and tokenisers, which handle input formatting and tokenisation internally. As these processes conflict with how transformer models understand syntax and semantics, no manual stemming, lemmatisation, or stopword removal was done. Final input handling—including padding, truncation, and conversion to input IDs—was done automatically by the model's built-in preprocessor.

3.3 Models used and evaluation pipeline

3.3.1 Classical ML models

A range of classical ML models was used to execute the binary classification task of identifying LLM-generated writings from human-authored ones. The chosen models represent many algorithmic families, facilitating a thorough assessment across distinct learning paradigms. The employed models are described in Table 3.2:

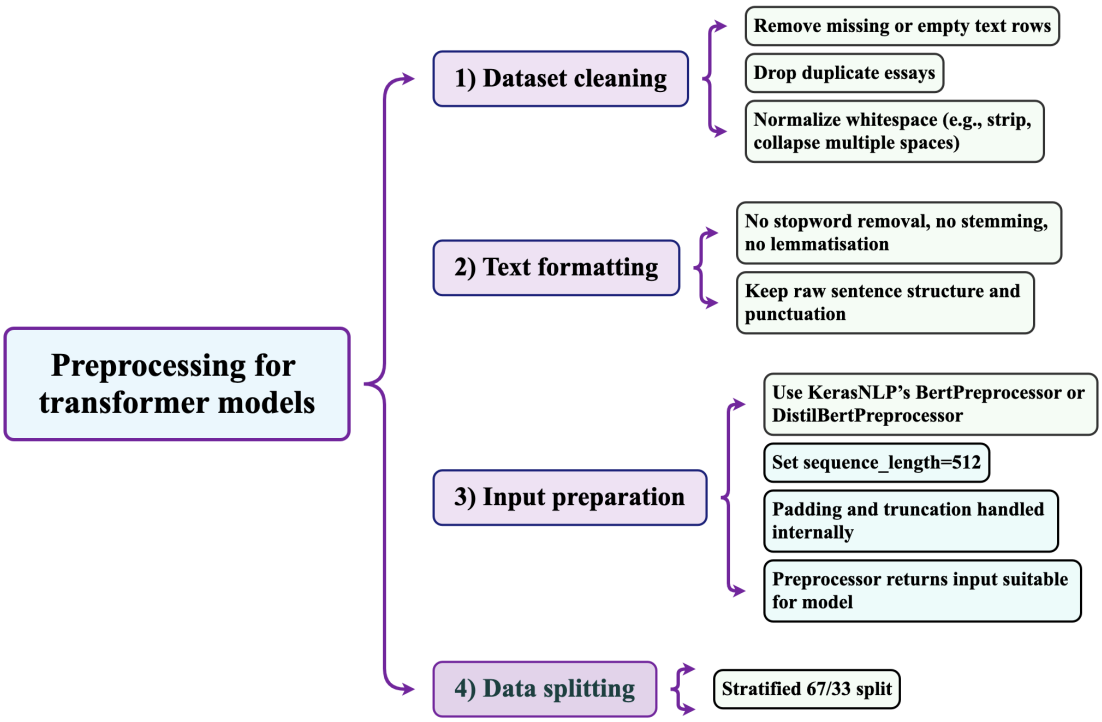


Figure 3.6: Preprocessing pipeline for transformer models

Table 3.2: Summary of classical ML models used in this study

Model	Description
Logistic Regression (LR)	A linear model commonly used for binary classification that predicts the probability of a sample belonging to a given class.
Support Vector Machine (SVM)	A margin-based classifier with a linear kernel that seeks to maximize the decision boundary between classes in high-dimensional space.
Decision Tree (DT)	A tree-structured model that partitions the feature space based on learned decision rules.
Random Forest (RF)	An ensemble of decision trees trained on random subsets to improve generalization.
K-Nearest Neighbors (KNN)	An instance-based algorithm that classifies samples based on the majority label among the nearest neighbors.
Multinomial Naive Bayes (NB)	A probabilistic model well-suited for text data, using Bayes' theorem with strong independence assumptions.
Multilayer Perceptron (MLP)	A simple neural network composed of fully connected layers for capturing non-linear relationships.

The incorporation of linear, tree-based, ensemble, instance-based, probabilistic, and basic neural models guarantees a comprehensive and varied comparison of traditional methods for AI-generated text identification.

3.3.2 Transformer models

A set of transformer-based models was used to address the classification problem of identifying AI-generated text in order to supplement the classical models. Pre-trained on large datasets, these models can be fine-tuned to fit particular classification needs. Fine-tuning allows the models to detect slight stylistic and semantic signals in the text separating LLM-generated results from human-authored ones. The transformer models employed in this work are summarised in Table 3.3:

Table 3.3: Summary of deep learning models used in this study

Model	**Description**
BERT	Bidirectional Encoder Representations from Transformers is a deep learning model that analyses text in both directions to understand context, facilitating superior performance across many NLP applications (Özkurt, 2024).
DistilBERT	DistilBERT is a compressed, accelerated variant of BERT that preserves the majority of BERT's linguistic comprehension capabilities while utilising fewer parameters, hence enhancing efficiency and suitability for resource-constrained contexts (Özkurt, 2024).
RoBERTa	RoBERTa enhances BERT by refining the training methodology, utilising increased data, extended training durations, and eliminating specific limitations to achieve superior accuracy (Özkurt, 2024).

3.3.3 Evaluation strategy

Standard binary classification metrics including accuracy, precision, recall, and F1-score were used to evaluate the performance of each model. All models had confusion matrices created to offer more understanding of classification behaviour. A robustness analysis was also done by measuring model performance across essay length categories mentioned in Table 3.4, helping to assess sensitivity to input length variations.

Table 3.4: Essay length categories used for performance evaluation

Category	Word count range
Short	$\leq$ 200 words
Medium	201–500 words
Long	> 500 words

This combined method of statistical measures and qualitative error analysis provides a complete picture of the performance and generalisation capacity of every model.

Chapter 4

Practical work and Results

4.1 Results for classical ML models

The results of the evaluation of classical ML models applied to the AI-generated text detection task are presented in this section, followed by a deeper analysis to explore model performance across different categories of text length.

4.1.1 Evaluation results for classical ML models

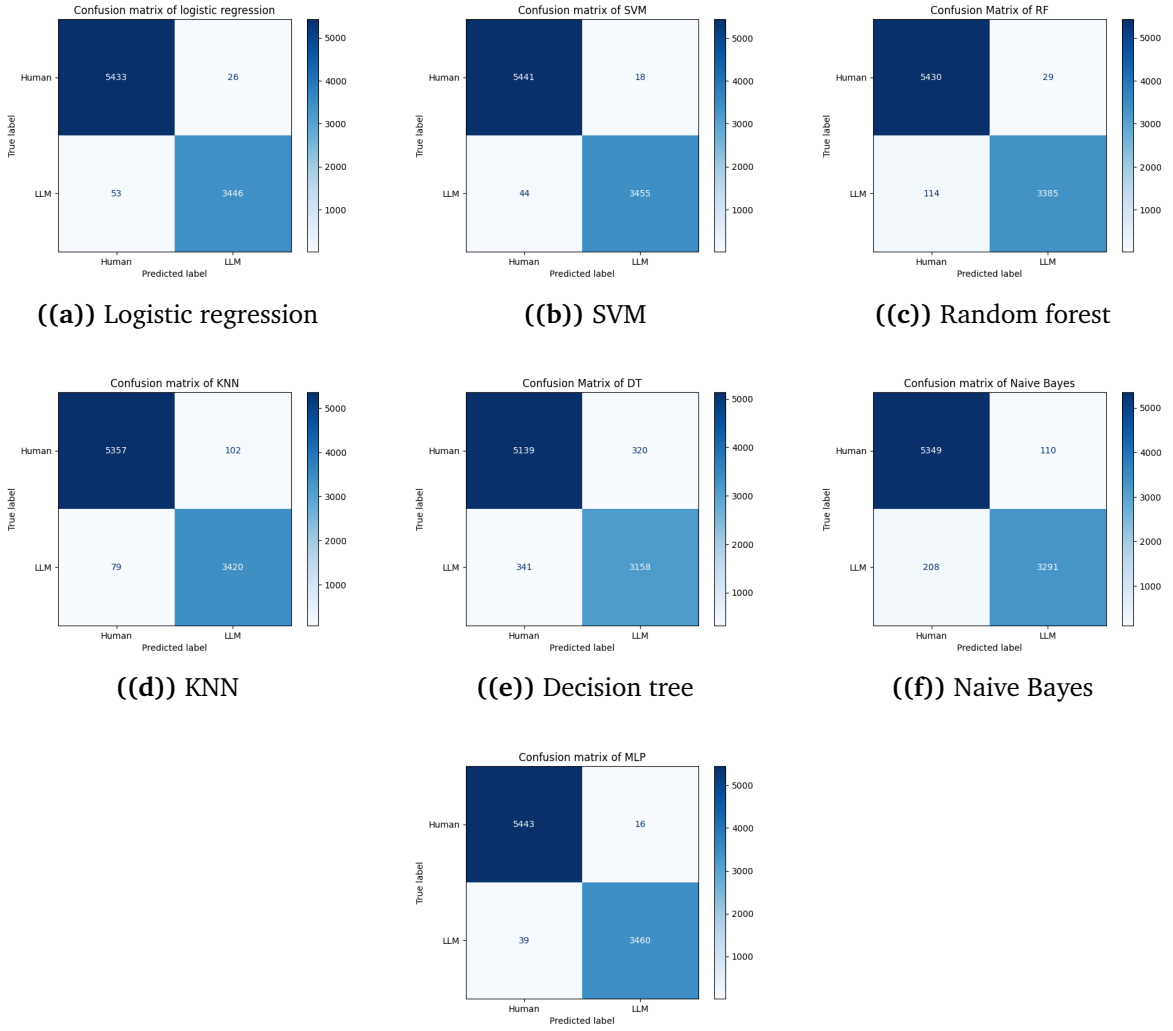
This section focuses on highlighting the performance of the traditional ML models in terms of accuracy, precision, recall, and F1-score for the LLM class.

Table 4.1: Performance comparison of classical ML models

Model	Accuracy	Precision	Recall	F1-score
LR	0.99	0.99	0.98	0.99
SVM	0.99	0.99	0.99	0.99
DT	0.93	0.91	0.90	0.91
RF	0.98	0.99	0.97	0.98
KNN	0.98	0.98	0.98	0.98
NB	0.96	0.96	0.94	0.95
MLP	0.99	0.99	0.99	0.99

The evaluation results for all classical machine learning models applied to the AI-generated text detection task are shown in Table 4.1. LR, SVM, and MLP achieved the best performance, with overall accuracies of 99% and F1-scores for LLM detection reaching 0.99. These models demonstrated very high precision and recall, highlighting strong abilities both to detect AI-generated texts and to avoid false positives. RF and KNN also performed well, achieving accuracies of 98%, although RF showed lower recall (0.97) for LLM texts. DT and NB exhibited lower performance, with DT suffering from overfitting and achieving an accuracy of 93%, and NB reaching 96% with a noticeable drop in LLM recall (0.94). These findings confirm that linear models (LR, SVM) and lightweight neural architectures (MLP) are particularly

well-suited for sparse, high-dimensional text feature spaces like those produced by TF-IDF with 10000 features when addressing AI-generated content detection.



((a)) Logistic regression

((b)) SVM

((c)) Random forest

((d)) KNN

((e)) Decision tree

((f)) Naive Bayes

((g)) MLP classifier

Figure 4.1: Confusion matrices for ML models used in this study.

Figure 4.1 presents the confusion matrices for all classical ML models evaluated in this study. As observed, LR, SVM, and MLP achieved highly accurate predictions for both classes, with minimal misclassification between Human and LLM texts. RF and KNN also performed well but exhibited slightly higher rates of misclassifying LLM texts as Human compared to linear models. DT and NB showed relatively larger confusion between classes, particularly DT, which misclassified a notable number of LLM samples as Human. Overall, the confusion matrices visually support the numerical results reported earlier, confirming that linear models and lightweight neural architectures are highly effective when addressing AI-generated text detection using the proposed methodology.

4.1.2 Performance analysis by essay length categories

To better understand how the classification models handle variations in input length, an additional analysis was conducted by grouping the test essays into three categories based on word count: Short ($\leq$ 200 words), Medium (201–500 words), and Long ($>$ 500 words). LR, SVM, and MLP classifiers were evaluated separately for each category. The goal was to assess whether text length affects the models' ability in classifying human and LLM-generated text.

Table 4.2: Performance of selected ML models across different essay length categories.

Model	Category	Accuracy	F1-score (LLM)
LR	Short	0.9891	0.9844
LR	Medium	0.9906	0.9900
LR	Long	0.9945	0.9633
SVM	Short	0.9891	0.9845
SVM	Medium	0.9926	0.9921
SVM	Long	0.9969	0.9793
MLP	Short	0.9891	0.9845
MLP	Medium	0.9938	0.9934
MLP	Long	0.9963	0.9752

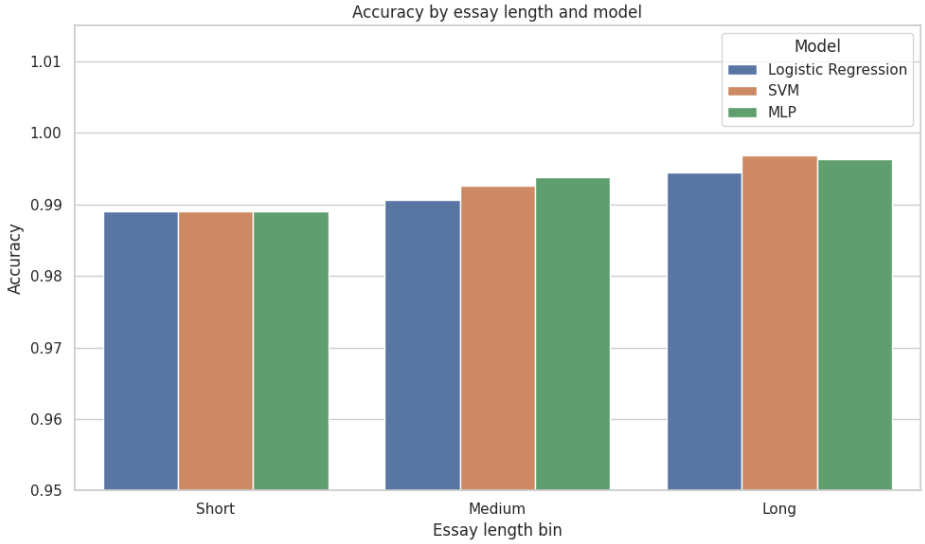


Figure 4.2: Accuracy performance of LR, SVM, and MLP models across different essay length categories.

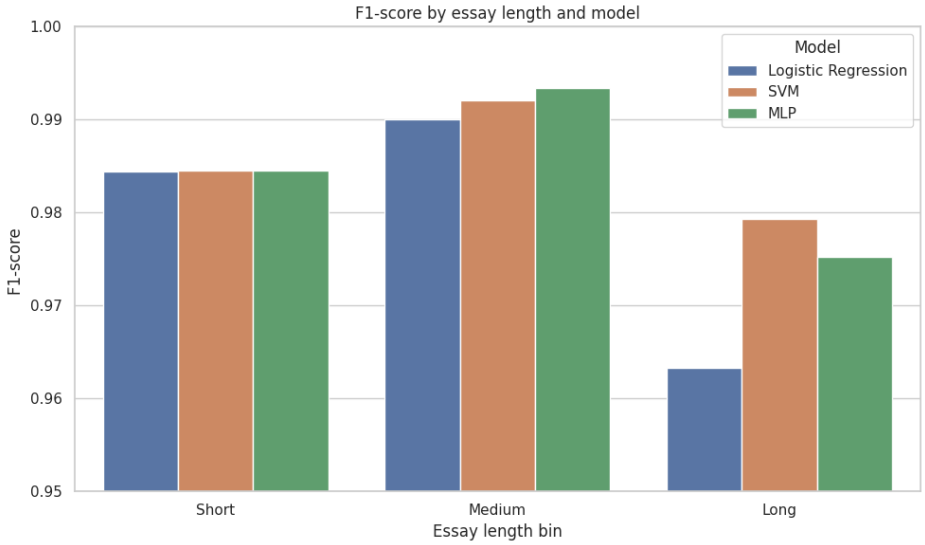


Figure 4.3: F1-score performance of LR, SVM, and MLP models across different essay length categories.

The findings in Table 4.2, Figure 4.2, and Figure 4.3 show that the all three models perform well in every category of length. Regardless of the essay length, overall accuracies stayed above 98.9%, suggesting strong robustness. The F1-scores showed little variation, especially for long essays where a small drop was seen, particularly for MLP and LR models. Longer essays' rising linguistic complexity and stylistic variety could explain this decline. The results, therefore, show that the models keep great generalisation capacity over various input lengths, which supports the reliability of the classification pipeline in practical uses.

4.2 Results for transformer models

4.2.1 BERT-based classification

We evaluated the performance of the BERT model across 5 configurations, each designed to assess the effect of fine-tuning, learning rate, and training duration on classifying human-written versus AI-generated text. Table 4.3 summarizes the training settings and outcomes of these experiments.

Table 4.3: Summary of training experiments

Experiment	Fine tuning	Epochs	lr[1]	Accuracy	Macro F1	F1 (0)	F1 (1)
Exp. 1	No	1	5e-4	**0.89**	**0.88**	0.91	0.86
Exp. 2	Yes	1	3e-5	0.60	0.55	0.70	0.41
Exp. 3	Yes	3	3e-5	0.32	0.28	0.11	0.46
Exp. 4	Yes	5	3e-5	0.32	0.28	0.11	0.46
Exp. 5	Yes	10 (ES@2)[2]	1e-5	0.32	0.28	0.11	0.46

In experiment 1, without non fine tuning, BERT achieved the best overall performance with an accuracy of **89%** and an F1-score of **0.88**. In contrast, fine-tuning the full BERT model in experiments 2-5 led to significant performance degradation. Despite varying the number of training epochs and reducing the learning rate, the model consistently over-fitted toward the LLM class. The Human class F1-score dropped to 0.11 in all fine-tuned cases, with overall accuracy dropping to 32%. The validation loss curves in Figure 4.4, showed little to no improvement across epochs, and early stopping in experiment 5 confirmed convergence to a suboptimal solution. These results indicate that the pre-trained BERT features were already effective and full fine-tuning negatively impacted generalization.

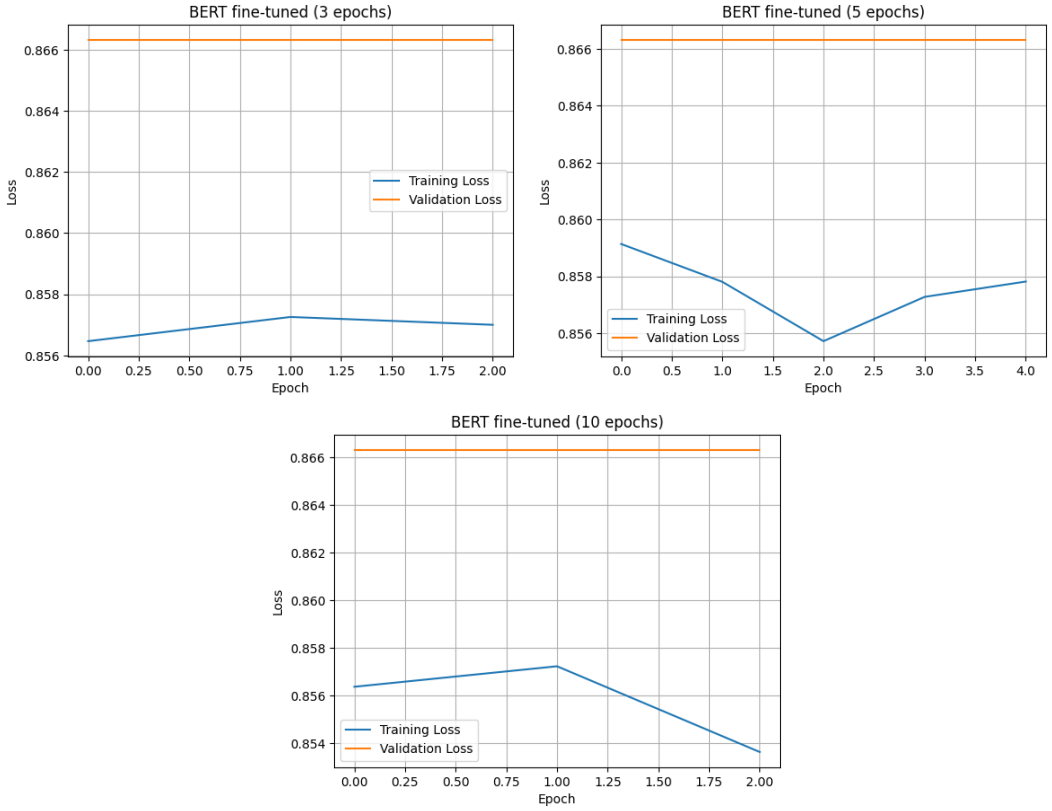


Figure 4.4: Training and validation loss across fine-tuned BERT experiments

4.2.2 DistilBERT-based classification

Like BERT, we evaluated DistilBERT's ability across 5 experimental configurations, with results summarized in Table 4.4. Without fine tuning, we got an accuracy of 88% and an F1-score of 0.88, this test performed the best. On the other hand, full fine-tuning was investigated in experiments 2 through 5. In experiment 2, accuracy dropped dramatically to 42% after just one fine-tuning epoch. The model continued to be stuck in a biassed state, over-predicting the LLM class, even with longer training

[1]learning rate
[2]Earlier stopping with *patience* = 2

Table 4.4: Summary of DistilBERT training experiments

Experiment	Fine tuning	Epochs	lr	Accuracy	Macro F1	F1 (Human)	F1 (LLM)
Exp. 1	No	1	5e-4	**0.88**	**0.88**	0.90	0.87
Exp. 2	Yes	1	3e-5	0.42	0.42	0.42	0.43
Exp. 3	Yes	3	3e-5	0.42	0.42	0.42	0.43
Exp. 4	Yes	5	3e-5	0.42	0.42	0.42	0.43
Exp. 5	Yes	10	1e-5	0.39	0.28	0.00	0.56

times (experiments 3 and 4). In experiment 5, a lower learning rate (1e-5) and early stopping led to model collapse, predicting nearly all samples as LLM-generated, with Human class recall at 0.00 and accuracy at 39%.

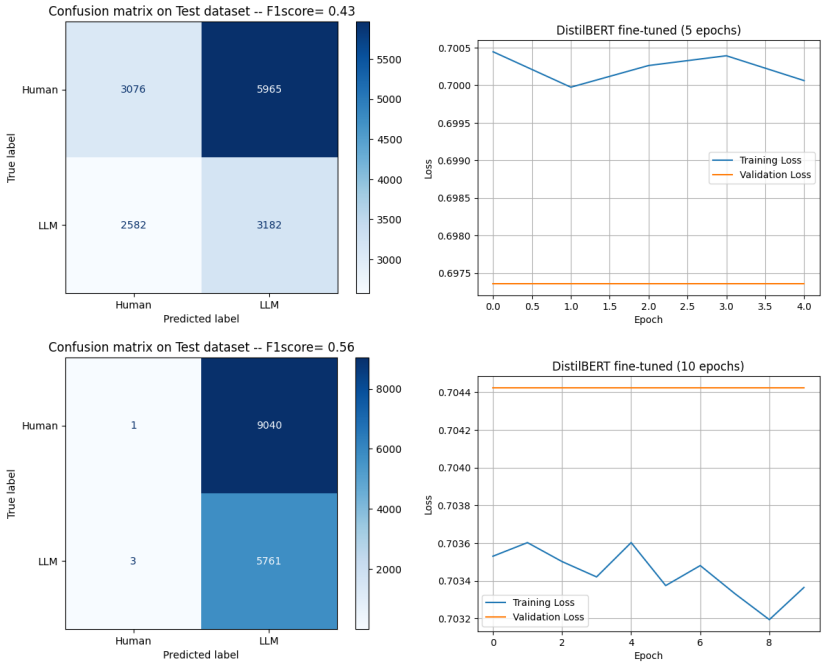


Figure 4.5: Training and validation loss, and confusion matrices for fine-tuned DistilBERT experiments

4.2.3 RoBERTa-based classification

Table 4.5 presents a summary of the results from various configurations tested with RoBERTa.

Table 4.5: Summary of RoBERTa training experiments

Experiment	Fine tuning	Epochs	LR	Accuracy	Macro F1	F1 (Human)	F1 (LLM)
Exp. 1	No	1	5e-4	**0.97**	**0.97**	0.97	0.96
Exp. 2	Yes	1	3e-5	0.61	0.38	0.76	0.00
Exp. 3	Yes	3	3e-5	0.61	0.38	0.76	0.00
Exp. 4	Yes	5	3e-5	0.61	0.38	0.76	0.00
Exp. 5	Yes	10	1e-5	0.61	0.38	0.76	0.00

Experiment 1 used RoBERTa without fine tuning. This yielded outstanding results, with an accuracy of 97% and macro F1-score of 0.97. Both human and LLM-generated texts were classified with very high precision and recall. Experiments 2–5 investigated full fine-tuning. In experiment 2, the model collapsed, predicting all texts as human-written, resulting in an F1 score of 0.00 for LLM detection and 61% accuracy, reflecting the dataset's human class proportion. Extended training (experiments 3–4) and a reduced learning rate with early stopping (experiment 5) failed to mitigate this bias, with all fine-tuned configurations remaining non-generalizing.

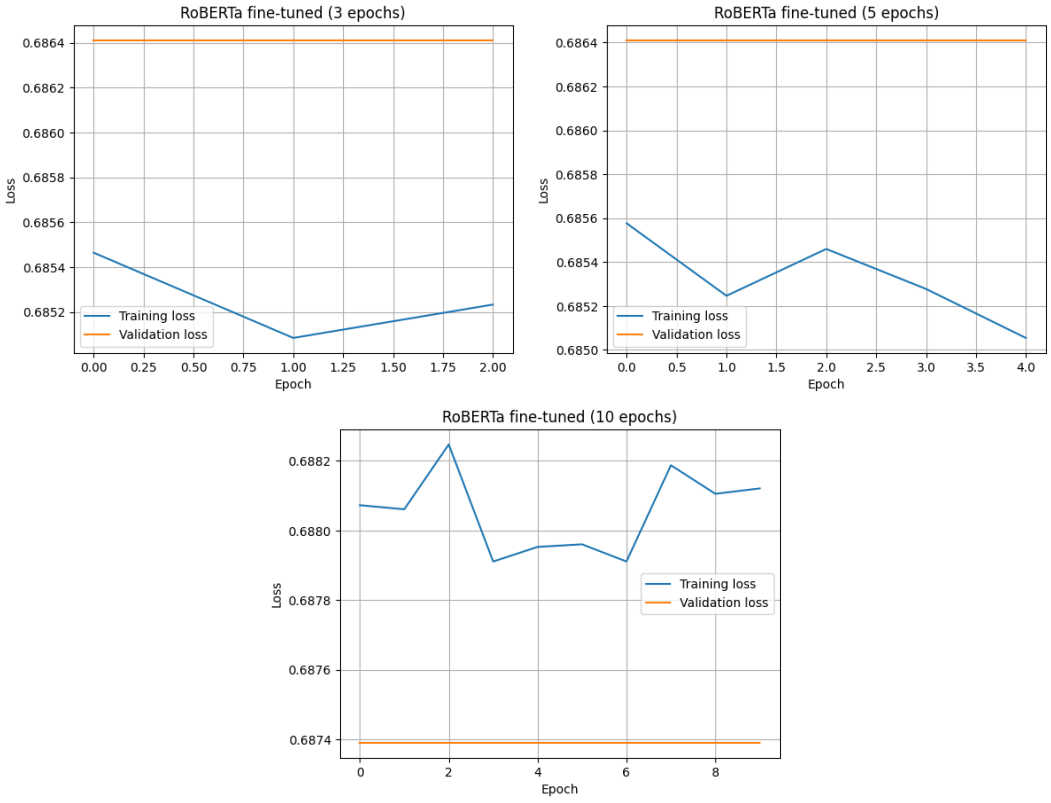


Figure 4.6: Training/validation loss curves for fine-tuned RoBERTa experiments

RoBERTa's frozen configuration outperformed all models, achieving near-perfect classification. However, its fine-tuning was unstable, confirming that freezing the backbone ensures the most reliable and accurate results for this task.

4.2.4 Comparative analysis across models

We evaluated BERT, DistilBERT, and RoBERTa under identical experimental conditions, with Table 4.6 summarizing each model's best-performing configuration.

Table 4.6: Best performance per model (frozen configurations)

Model	Accuracy	Macro F1	F1 (LLM)
BERT	0.89	0.88	0.86
DistilBERT	0.88	0.88	0.87
RoBERTa	0.97	0.97	0.96

RoBERTa achieved the highest scores in its frozen configuration.

4.2.5 Performance by essay length: transformer models

We assessed frozen BERT, DistilBERT, and RoBERTa on short (≤ 200), medium (201–500), and long (> 500) essays, focusing on accuracy and LLM F1-score (Figures 4.7, 4.8). RoBERTa outperformed others across all essay lengths, achieving the highest F1-score and accuracy. All models performed robustly on short and medium essays but showed significant F1-score decrease for long essays, particularly BERT and DistilBERT. Frozen transformers excel with shorter texts but struggle with long essays, with RoBERTa demonstrating superior robustness in capturing long-range dependencies.

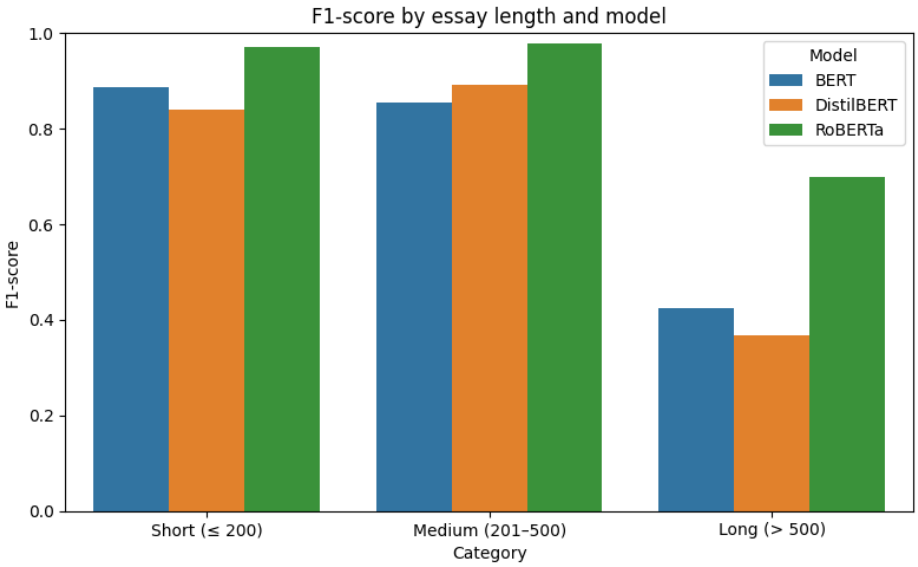


Figure 4.7: F1-score performance of BERT, DistilBERT, and RoBERTa across different essay length categories.

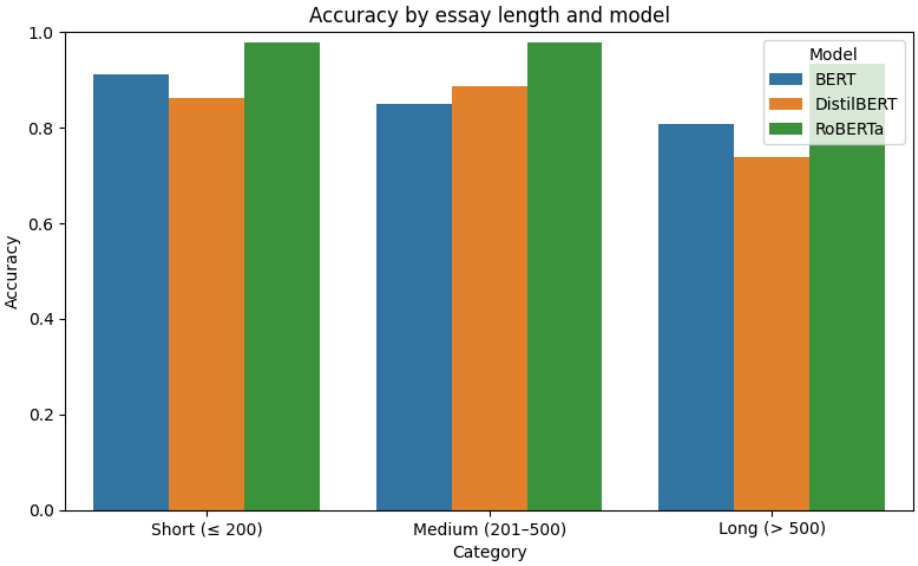


Figure 4.8: Accuracy performance of BERT, DistilBERT, and RoBERTa across different essay length categories.

Chapter 5

Discussion and critical analysis

5.1 What worked best and why

Among the classical models, the most efficient ones in this research were LR, SVM, and MLP; among the transformer-based models, RoBERTa. Because of their compatibility with high-dimensional TF-IDF feature representations, classical models reached F1-scores as high as 0.99. In addition, class weighting rather than resampling helped this performance even more, a strategy advised for moderate class imbalance situations to prevent overfitting or information loss. Among transformer models, RoBERTa achieved the highest scores, with 97% accuracy and F1-score of 0.96. This can be attributed to RoBERTa's optimized training regime, which improves upon BERT by removing the Next Sentence Prediction (NSP) task and using dynamic masking and a larger corpus (Al-Jarrah and Al-Hammouri, 2020).

5.2 Comparison to benchmarks and critical analysis

This work uses a simpler but varied pipeline, combining seven classical machine learning models with three transformer-based architectures (BERT, DistilBERT, RoBERTa) and applies them to a concatenated and cleaned dataset of around 46000 samples when compared to benchmarks such as DetectRL (Wu, Zhan, Wong, Yang, Yang, Yuan and Chao, 2024), FastDetectGPT (Wu, Zhan, Wong, Yang, Liu, Chao and Zhang, 2024), and GPTDetector Dou et al. (n.d.). DetectRL used over 100000 human and 134000 LLM-generated samples from several domains and adversarial situations, GPTDetector assessed on actual AI-generated academic texts. In contrast, (Yadagiri and Pakray, 2025) used only a small subset of the dataset we used and did not fully explore its diversity, which limits the generalizability of their findings. In addition, the extended dataset we use provides more coverage and more support for robustness testing across input variation. Unlike authors (Yadagiri and Pakray, 2025), who studied fully fine-tuned transformers, we found fine-tuning caused performance drops and model instability. Our frozen RoBERTa configuration (F1 = 0.97) outperformed fine-tuned models, aligning with studies favouring minimal adaptation of pre-trained models in low-resource settings.

Our evaluation emphasised text variability via length-based bins (short, medium, long) in comparison to Wu et al.(Wu, Zhan, Wong, Yang, Yang, Yuan and Chao, 2024), who tested detectors under real-world disturbances including paraphrasing and spelling noise. Although this is less thorough in adversarial coverage, it provides a fresh viewpoint on classifier robustness, especially pertinent in educational and editorial contexts where essay length relates to writing complexity. Though he did not test against baseline classical models, Dou (2022) (**?**) suggested a GPTDetector framework using likelihood and linguistic divergence measures. On the other hand, our work offers a more understandable comparison over transformer-based architectures and classical ML classifiers. In our work, adversarial augmentations were not used, which is a drawback of our work. To increase generalisability, future research should include multilingual or domain-specific material as well as adversarial transformations, e.g., paraphrasing or grammatical noise. Including technologies such as DetectGPT or ChatGPTZero into our assessment would also help to create more robust comparison baselines. This work shows that when trained on well-preprocessed input, LR and SVM, two simpler architectures, can equal or outperform the performance of bigger, finely tuned models. Our thorough assessment of classical and deep models, robustness by essay length, and fine-tuning stability analysis significantly advanced our understanding of developing scalable and interpretable LLM-detection systems.

5.3 Legal, social, ethical and professional issues in ML-based AI-generated text detection

The rapid progress of generative AI models like GPT-3 and GPT-4 complicates detecting AI-generated content. Meanwhile, machine learning-based detection systems raise legal, social, ethical, and professional concerns, requiring careful management to ensure transparency, fairness, and responsible use.

Table 5.1: Summary of legal, social, ethical, and professional issues in AI-generated text detection

Issue type	Key concerns
Legal issues	- Privacy risks from processing user data without consent, potentially violating GDPR (Ghiurău and Popescu, 2025). - Misclassification of human-authored content can infringe intellectual property rights. - Legal frameworks such as the EU AI Act and Digital Services Act require transparency and fairness in detection systems (Ghiurău and Popescu, 2025).
Social issues	- Detection tools may create mistrust if users feel constantly monitored or wrongly flagged (Valiaiev, 2023). - Unequal access to detection technologies could reinforce inequalities in education, journalism, and the workplace (Ghiurău and Popescu, 2025; Valiaiev, 2023). - Potential impact on freedom of expression and social media moderation policies (Ghiurău and Popescu, 2025).
Ethical issues	- False positives can harm reputations, lead to academic or professional sanctions (Valiaiev, 2023). - Biased models may disproportionately affect non-native speakers and minority groups (Ghiurău and Popescu, 2025). - The black-box nature of ML models reduces accountability and user understanding (Valiaiev, 2023; Ghiurău and Popescu, 2025). - Risk of overreach threatening freedom of speech in academic or journalistic contexts (Valiaiev, 2023).
Professional issues	- Scientists and developers must prioritise fairness, transparency, explainability, and accountability (Valiaiev, 2023). - Ethical responsibility to communicate system limitations clearly to end-users (Ghiurău and Popescu, 2025). - Need for interdisciplinary collaboration to guide the development and governance of detection tools (Valiaiev, 2023).

As summarised in Table 5.1, the implementation of ML-based detection systems for content authenticity requires a careful balance between technological capabilities, ethical principles, legal compliance, and the protection of individual rights.

Chapter 6

Conclusion

This study evaluated classical ML and transformer-based models for detecting AI-generated text using a diverse, expanded dataset. Seven classical models and three fine-tuned transformer models were assessed for classification performance and robustness across essay lengths. The study also addressed legal and ethical considerations, emphasizing fairness, transparency, and responsible detection practices. These findings establish a practical framework for benchmarking and enhancing AI-generated text detection systems.

Appendix: Code structure and execution guide

This coursework is implemented in two notebooks, each corresponding to a different family of models:

- **Notebook 1** implements all experiments related to classical machine learning models using TF-IDF features.
- **Notebook 2** contains the experiments using transformer-based models (BERT, DistilBERT, and RoBERTa), evaluated under both fine-tuned and frozen configurations.

Each notebook includes its own preprocessing pipeline tailored to the model type. To enhance clarity and reproducibility, both preprocessing and experimentation are contained within their respective notebooks.
The final dataset used in this study is created by concatenating two source datasets, both of which are included in the submitted ZIP file. The merging and preparation steps are integrated into the notebooks, so no external preprocessing is required.
All notebooks were tested in standard Python environments (e.g., Jupyter or Google Colab) using common libraries such as `scikit-learn`, `pandas`, etc.

References

Al-Jarrah, H. and Al-Hammouri, R. (2020), 'Hr@just team at semeval-2020 task 4: The impact of roberta transformer for evaluation common sense understanding'.
URL: *https://github.com/wangcunxiang/SemEval2020-Task4-Commonsense-Validation-and-Explanation* pages 32

Anderson, N., Belavy, D. L., Perle, S. M., Hendricks, S., Hespanhol, L., Verhagen, E. and Memon, A. R. (2023), 'Ai did not write this manuscript, or did it? can we trick the ai text detector into generated texts? the potential future of chatgpt and ai in sports exercise medicine manuscript generation', *BMJ Open Sport and Exercise Medicine* **9**. pages 8

Bao, G., Zhao, Y., Teng, Z., Yang, L. and Zhang, Y. (2023), 'Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature'.
URL: *http://arxiv.org/abs/2310.05130* pages 14

Buda, M., Maki, A. and Mazurowski, M. A. (2018), 'A systematic study of the class imbalance problem in convolutional neural networks', *Neural Networks* **106**, 249–259. pages 20

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S. and Smith, N. A. (n.d.), 'Human evaluation of generated text'.
URL: *www.nltk.org/* pages 12

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N. A., Choi, Y. and Allen, P. G. (n.d.), 'Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text'. pages 13, 14, 15, 32

Dugan, L., Ippolito, D., Kirubarajan, A., Shi, S. and Callison-Burch, C. (2023), 'Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text'.
URL: *https://github.com/liamdugan/human-detection* pages 14

Elkhatat, A. M., Elsaid, K. and Almeer, S. (2023), 'Evaluating the efficacy of ai content detection tools in differentiating between human and ai-generated text', *International Journal for Educational Integrity* **19**. pages 8, 9, 11

Feurer, M., Eggensperger, K., Bergman, E., Pfisterer, F., Bischl, B. and Hutter, F. (2022), 'Mind the gap: Measuring generalization performance across multiple

objectives'.
URL: *http://arxiv.org/abs/2212.04183 http://dx.doi.org/10.1007/978-3-031-30047-9*$_1$1 *pages* 12

Gallé, M., Rozen, J., Kruszewski, G. and Elsahar, H. (2021), 'Unsupervised and distributional detection of machine-generated text'.
URL: *http://arxiv.org/abs/2111.02878* pages 14

Gehman, S., Gururangan, S., Sap, M., Choi, Y. and Smith, N. A. (2020), 'Realtoxicityprompts: Evaluating neural toxic degeneration in language models'.
URL: *http://arxiv.org/abs/2009.11462* pages 12

Ghiurău, D. and Popescu, D. E. (2025), 'Distinguishing reality from ai: Approaches for detecting synthetic content', *Computers* **14**. pages 34

Gu, C., Huang, C., Zheng, X., Chang, K.-W. and Hsieh, C.-J. (2022), 'Watermarking pre-trained language models with backdooring'.
URL: *http://arxiv.org/abs/2210.07543* pages 14

Guo, Z. and Yu, S. (2023), 'Authentigpt: Detecting machine-generated text via black-box language models denoising'.
URL: *http://arxiv.org/abs/2311.07700* pages 14

Hamed, A. A. and Wu, X. (2023), 'Detection of chatgpt fake science with the xfakesci learning algorithm'.
URL: *http://arxiv.org/abs/2308.11767* pages 13, 14, 15

He, X., Shen, X., Chen, Z., Backes, M. and Zhang, Y. (2023), 'Mgtbench: Benchmarking machine-generated text detection'.
URL: *http://arxiv.org/abs/2303.14822* pages 11, 18

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A. and Fung, P. (2023), 'Survey of hallucination in natural language generation'. pages 11, 12

Koike, R., Kaneko, M. and Okazaki, N. (2023), 'Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples'.
URL: *http://arxiv.org/abs/2307.11729* pages 14

Mindner, L., Schlippe, T. and Schaaff, K. (2023), 'Classification of human- and ai-generated texts: Investigating features for chatgpt'.
URL: *http://arxiv.org/abs/2308.05341 http://dx.doi.org/10.1007/978-981-99-7947-9*$_1$2 *pages* 13, 14, 15

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D. and Finn, C. (2023), 'Detectgpt: Zero-shot machine-generated text detection using probability curvature'.
URL: *http://arxiv.org/abs/2301.11305* pages 13, 14, 15

Shah, A., Ranka, P., Dedhia, U., Prasad, S., Muni, S. and Bhowmick, K. (n.d.), 'Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features', *IJACSA) International Journal of Advanced Computer Science and Applications* **14**, 2023.
URL: *www.ijacsa.thesai.org* pages 13, 14, 15

Shi, Z., Wang, Y., Yin, F., Chen, X., Chang, K.-W. and Hsieh, C.-J. (2023), 'Red teaming language model detectors with language models'.
URL: *http://arxiv.org/abs/2305.19713* pages 11

Solaiman, I., Brundage, M., Jack, O., Openai, C., Openai, A. A., Herbert-Voss, A., Openai, J. W., Openai, A. R., Openai, G. K., Wook, J., Openai, K., Kreps, S., Politiwatch, M. M., Newhouse, A., Blazakis, J., Mcguffie, K. and Wang, J. (2019), 'Openai report release strategies and the social impacts of language models'. pages 14

Uchendu, A., Lee, J., Shen, H., Le, T., Huang, T.-H. . K. and Lee, D. (2023), 'Does human collaboration enhance the accuracy of identifying llm-generated deepfake texts?'.
URL: *www.aaai.org* pages 14

Valiaiev, D. (2023), 'Detection of machine-generated text: Literature survey', *ACM* .
URL: *https://doi.org/0000001.0000001* pages 34

Wang, H., Li, J. and Li, Z. (n.d.), 'Ai-generated text detection and classification based on bert deep learning algorithm'. pages 8, 13, 15

Wu, J., Yang, S., Zhan, R., Yuan, Y., Chao, L. S. and Wong, D. F. (2025), 'A survey on llm-generated text detection: Necessity, methods, and future directions under a creative commons attribution-noncommercial-noderivatives 4.0 international (cc by-nc-nd 4.0) license'.
URL: *https://doi.org/10.1162/colia00549* pages 6, 8, 9, 11, 12, 18

Wu, J., Zhan, R., Wong, D. F., Yang, S., Liu, X., Chao, L. S. and Zhang, M. (2024), 'Who wrote this? the key to zero-shot llm-generated text detection is gecscore'.
URL: *http://arxiv.org/abs/2405.04286* pages 14, 32

Wu, J., Zhan, R., Wong, D. F., Yang, S., Yang, X., Yuan, Y. and Chao, L. S. (2024), 'Detectrl: Benchmarking llm-generated text detection in real-world scenarios'.
URL: *http://arxiv.org/abs/2410.23746* pages 32, 33

Yadagiri, A. and Pakray, P. (2025), Deep learning strategies for identifying machine-generated text, *in* 'Proceedings of the 2025 19th International Conference on Ubiquitous Information Management and Communication, IMCOM 2025', Institute of Electrical and Electronics Engineers Inc. pages 8, 13, 15, 32

Yu, X., Qi, Y., Chen, K., Chen, G., Yang, X., Zhu, P., Shang, X., Zhang, W. and Yu, N. (2023), 'Dpic: Decoupling prompt and intrinsic characteristics for llm generated text detection'.
URL: *http://arxiv.org/abs/2305.12519* pages 14

Özkurt, C. (2024), 'Comparative analysis of state-of-the-art q a models: Bert, roberta, distilbert, and albert on squad v2 dataset', *Chaos and Fractals* . pages 22