Detecting AI-Created Essays using Artificial Intelligence

Vasyl Dykun
Department of Computer and Software Engineering
Technological University of the Shannon
Athlone, Ireland
a00315339@student.tus.ie

Liam Shanley
Department of Computer and Software Engineering
Technological University of the Shannon
Athlone, Ireland
a00291593@student.tus.ie

***Abstract*—This paper aims to address the problem of AI-enabled plagiarism, which has been facilitated by the quickly-growing sophistication of large language models (LLMs). A random forest model was trained on three distinct datasets. Each trained model was tested on unseen, unrelated datasets. The best model was determined using various performance metrics, including accuracy, F1-score and (low) false positive rate (FPR). The limitations of the best model are defined and articulated in terms of FPR. Avenues for future research are identified and include exploration of alternative algorithms, data improvement and model optimization strategies.**

I. Introduction (*Heading 1*)

The growing capabilities of Large Language Models (LLMs) have precipitated a concurrent concern in a range of sectors. LLMs have been implicated in instances of misinformation, spam, fake news, gender bias and social harm [1]. In higher education, the issue of academic integrity is one which continually finds priority in discussions of LLMs [17]. The purpose of this project is to build a technological product to distinguish between student essays which have been written by students and those which have been machine-generated.

The report which follows attempts to accomplish the following objectives. First, a background to the use case is provided; this will outline the challenges which have emerged in academia and higher education since the introduction of ChatGPT. Also, there is a brief overview of the technical innovations which have enabled the strong performance of ChatGPT. The value of the project is then established and articulated through concurrent consideration of the business and technical contexts. Secondly, there is a review of the technology applied to AI-generated-text classification. A diversity of domains is included in this section, such as misinformation and fake news. This section concludes by highlighting the potential of zero-shot and different training methods, when employed in a black-box context. Thirdly, the review considers debates surrounding the feasibility of text detectors in light of the increasing sophistication of LLMs.

Through consulting available research on the design of LLMs, it is argued that the problem of text detection is a solvable one. The final section of the literature review considers the application of AI to the specific domain of AI-enabled plagiarism by students. The review succeeds in uncovering important aspects of the research process; these inform recommendations for product design. Resources available to the end user, such as data availability and computational resources, are flagged as issues for consideration during the product-design phase. Feature engineering using TFIDF is defined as an appropriate initial avenue towards addressing the research question.

II. Literature Review

A. Background

This project aims to address the problem of academic integrity. The section which follows contemplates the context in which this problem has come to be emphasized. This context is considered to have two main aspects: the business context and the technical context. Business context is concerned with the business problem of the end user. Efforts are made to answer the following question: what issues have manifested in academia and higher education following the introduction of ChatGPT? In answering this, the prevalence of cheating by students and interaction effects between ChatGPT and cheating are highlighted. The technical context outlines technical innovations which have enabled students to cheat. This section outlines the specifications of ChatGPT and the influence of transformer architecture. Finally, the performance of more recent versions of ChatGPT is outlined.

1) Use Context

A rapid literature review by Chung found two primary issues concerning the use of ChatGPT in higher education: first, reliability of the information provided; secondly, the issue of academic honesty [1]. Regarding academic honesty, Cotton, Cotton & Shipway define two modes of presentation. The first was the possibility of plagiarism; the second, the unfair advantage enjoyed by those who have access to ChatGPT compared to those who do not. Strikingly, illustrating the adeptness of ChatGPT at producing human-like text, most of the cited article by Cotton, Cotton & Shipway was written by ChatGPT. Only suitable prompts and suitable structuring were provided by the authors [2]. Elsewhere, the sophistication of LLMs has resulted in AI-generated essays being considered highly original by existing plagiarism detectors [1].

Efficacy in evasion has had repercussions for the pre-existing problem of student cheating. Prior to the introduction of ChatGPT, research suggested that somewhere between 14% and 22% students engage in plagiarism, at some point – the higher figure coming from an Austrian study where students (assured of their anonymity) *admitted* to plagiarism [2]. Additionally, students who use ChatGPT are more likely to engage in plagiarism than those who don't [1]. Given the speed at which ChatGPT has been downloaded, it seems plausible that many students in higher education will also have access to this technology. When viewed in conjunction with

the cited evidence, an increasing rate of plagiarism in higher education would seem plausible into the future.

2) Technical Context

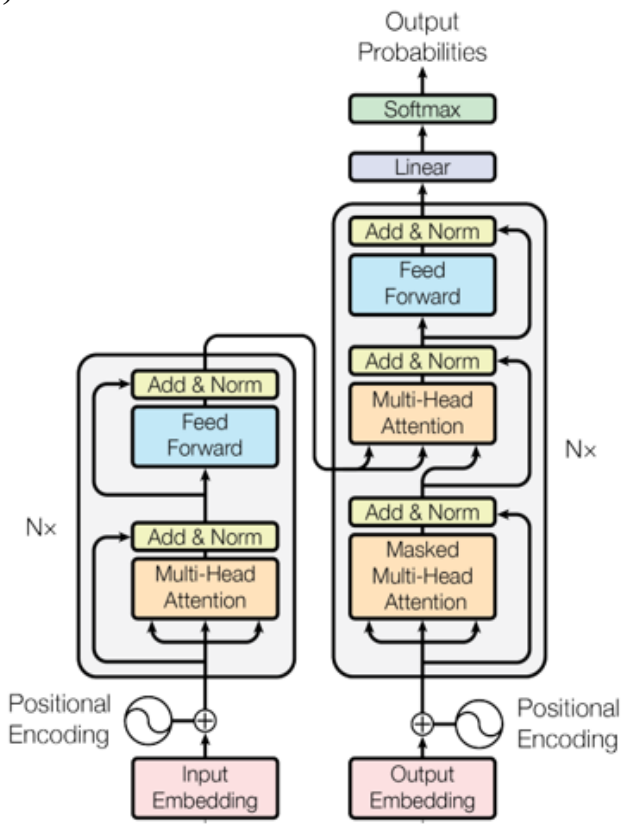


Figure 1: The Transformer - model architecture [20].

The most powerful LLMs, including ChatGPT, usually adopt the decoder-only transformer architecture (Figure 1). These models have tens to hundreds of billions of parameters, are tuned to human preferences and are trained on large volumes of text [3]. The first models based on this architecture were trained on 4.5 million sentence pairs with a vocabulary of 32,000 tokens [4]. GPT-3, on the other hand, was trained on 410 billion tokens. This illustrates the striking rate at which LLMs have been advancing since 2017. Similarly, GPT-3 has 175 billion parameters, which is 10 times more than any previous non-sparse language model. Improving specifications have resulted in the capacity of GPT-3 to generate text which human evaluators have difficulty distinguishing from human-written text [5]. Likewise, empirical studies on the zero-shot learning capabilities of GPT-3 have demonstrated the strength of GPT-3's performance on a diversity of NLP tasks. Reasoning and dialogue tasks have been remarked as particularly impressive [6]. The strong performances of LLMs on a range of tasks emphasizes the potential for their generalizability with further iterations.

GPT-4 was released on 13th March 2023 [7]. This model has shown human-level performance on various tasks. Additionally, showing its capability, this model passed a simulated bar exam, scoring in the top 10% of test takers [8]. The architecture of GPT-4 has not been made public, due to its replicability; however, it has been reported that GPT-4 was trained on 13 trillion tokens, making it 10 times larger in scale than GPT-3. Additionally, GPT-4 is reported to have 1.8 trillion parameters [9].

3) Potential Implications of GPT-4 in Context

Considering the challenges which have already been precipitated by GPT-3, it seems inevitable that a more powerful model will have dramatic implications for the field of academia and higher education. The following issues have been highlighted previously: reliability of information and the possibility of plagiarism [1]. GPT-4 seems to have made progress on the issue of reliability [8]. However, the improved performance of GPT-4 will likely worsen the already-challenging issues surrounding plagiarism.

The introduction of GPT-4 is likely to influence the possibility of plagiarism in numerous ways. First, where there is an unfair advantage associated with access to ChatGPT, an improved product will only increase this disparity. Secondly, the possibility of plagiarism is likely to increase with the introduction of more sophisticated models. Models with improved performance are likely to be better at evading detection. Consequently, the possibility of detection becomes less effective, when improved models are used. As a result, students who may have been deterred by the possibility of detection previously may no longer be deterred. This phenomenon could lead to an increase in the proportion of students engaged in cheating. This risk carries secondary risks: where a greater number of students engage in plagiarism, it may become more normatively acceptable to cheat. Significantly, the normalization of cheating in the culture of higher education may have the potential to undermine the objectives of the sector.

It is from this point that the value of detection is derived. The risks to the strategic objectives of academia and higher education can be addressed through the development of more sophisticated detection tools. Such tools have the potential to form some part of a broader strategy to mitigate against the risks posed by plagiarism.

B. Approaches to Text Classification

The problem of detecting AI-created essays falls within the broader research area of text classification. To date, research in this area has focused on issues such as fake news, gender bias, academic dishonesty, cybersecurity and social harm [1], [2], all of which have been influenced by the increasing sophistication of LLMs. The purpose of this section is to outline the main approaches to the problem of text identification and the methods employed to do so. It is intended that this process will identify avenues by which the research question may be addressed.

The level of access to LLM output logs has tended to determine how researchers approach the problem of machine-generated text detection. [1]. White-box approaches entail access to the model; this access is absent in black-box approaches and developers only enjoy API-level access to LLMs [2]. Within these approaches, research has been grouped into three methods: training, zero-shot and watermarking methods [2].

Training methods include either traditional machine learning or the tuning of pre-trained LLMs on binary data, consisting of human and machine-generated text. Zero-shot

methods leverage the properties of LLMs, such as probability distribution curves to perform self-detection. Watermarking involves placing information into the generated text which can be identified later [1].

1) White-Box Approaches

a) Training Methods

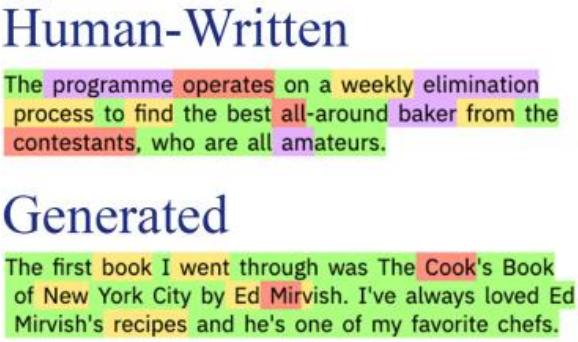


Figure 2: Top-k overlay within GLTR [3].

The Giant Language Model Test Room (GLTR) has full access to the language model distribution. Compared to GPT-2 or BARD, human language takes from the tails of the probability distribution more frequently. From this, GLTR employs statistical detection methods to examine the distributional qualities of the language to classify a piece of text [3]. The above tests were applied: (Test 1) the probability of the word, (Test 2) the absolute rank of the word and (Test 3) the entropy of the predicted distribution. Using colour codes, the model visualizes outliers based on the probability function, thereby assisting correct differentiation by humans (Figure 2). Unaided human performance was taken as a benchmark and correct classification, the performance metric. Using colour coding, subjects correctly differentiated the text sample in 72% of cases, compared to 54% without the tool [3].

b) Zero-shot methods

DetectGPT was designed to detect machine-generated fake news articles by harnessing an LLM's probability function [4]. Researchers had partial access to the source code and all experiments were all conducted in a white-box context [4].

Using the source model, the LLM attempted to detect its own samples through the examination of the log probability function. Text sampled from an LLM tended to occupy negative curvature regions of the model's log probability function; therefore, using this feature, differentiation between human- and machine-generated text is possible [4]. Data for human-generated text was taken from six different datasets. Machine-generated text was generated through prompting and all experiments involved balanced datasets. Receiver operating characteristic curve (AUROC) was selected as a performance metric. DetectGPT outperformed other models except for a LogRank model, which performed equally on one dataset. GPT-3 performs slightly better than Jurassic-2, achieving an AUROC of between 0.84 and 0.87 when tested on the datasets. It is noted that this performance is on par with supervised learning methods [4].

c) Watermarking methods

Recently, attention has been given to watermarking in contexts where access to the LLM has been granted at detection time. Here optimal statistical testing can be effective in identifying machine-generated text. Moreover, multi-bit watermarking can assist in identifying which model generated the text [5].

2) Black-Box Approaches

a) Training Methods

RoBERTa-MPU and BERT-MPU were designed to detect smaller fragments of text. Here, the text detection problem was approached as a 'Partial Positive Unlabelled' problem [6]. Two datasets were utilized: one of fake twitter tweets and one of short sentences, some in English and some in Chinese. Summary statistics informed the assumption that the text fragments in the datasets equate to the language of instant messaging or microblogging applications.

BERT and RoBeRTa were adopted to apply the novel Multiscale Positive Unlabelled (MPU) method. On the TweepFake dataset, scoring an accuracy of 91.4%, the RoBERTa-MPU model outperformed other RoBERTa baseline models, including those requiring additional finetuning [6].

On the English and Chinese datasets, several existing detectors were selected as benchmarks. On the English texts, based on F1 scores, BERT-MPU performs better than all other models on longer texts: it returns an F1 score of 96.8%±0.52%. However, RoBERTa-MPU performs best on sentence-length text: it returns an F1 score of 85.31%±1.8%. On sentence-length data RoBERTa-MPU improves on the best non-MPU model by 4%.

On Chinese sentences, RoBERTa-MPU shows a superior F1 score on both sentence-length and longer text: 97.42%±0.24% on full text and 89.37%±1.94% on sentences. RoBERTa-MPU shows an F1-score improvement of 1 percentage point and 5 percentage points for long and short texts, respectively, compared with the next-best model [6].

The application of the MPU models may contribute to tackling AI-enabled plagiarism in numerous respects. Most pertinently, the capability of MPU models to detect based on snippets, rather than entire LLM-generated essays. The sentence-level performance of this model appears promising. Furthermore, issues of inferior performance on non-English text have been raised previously. Therefore, the model's generalizability to other languages is promising avenue for future work.

b) Traditional classification methods

Fröhling and Zubiaga aimed to create an inexpensive first line of defense against the risks of LLMs. Features were derived from empirically observed differences between human- and machine-generated text. These included: counts of named entities per sentence; conference-cluster-related features; word-sentiment correlation; and simple counts of stylistic characteristics [7].

Engineered features were inputted into a supervised binary classification model and AUROC was chosen as the evaluation metric. Multiple classification algorithms are used, including Random Forest, Neural Networks (NN), Logistic Regression (LR) and Support Vector Machines. The resulting model achieved scores of 0.78 and 0.86 for accuracy and AUROC, respectively. However, the model is less effective in generalizing across LLMs. Finally, an ensemble model was constructed using LR and NN models; this improved the performance of the feature-based model by 0.01. Although the

superior performance of RoBERTa models was acknowledged, this model was considered competitive with LLM-trained detectors, since the contemporaneous LLM-based studies restricted their samples to a fixed length of 510 tokens, thereby, it is argued, accuracy may be inflated [7].

Traditional methods still seem promising: compared to more sophisticated LLM-based models, they are relatively inexpensive (both in terms of computational and human resources) to build and run. Furthermore, if the performance is comparable to that of an LLM, it is probable that a cost-benefit analysis would return a recommendation for the traditional methods.

c) Zero-shot methods

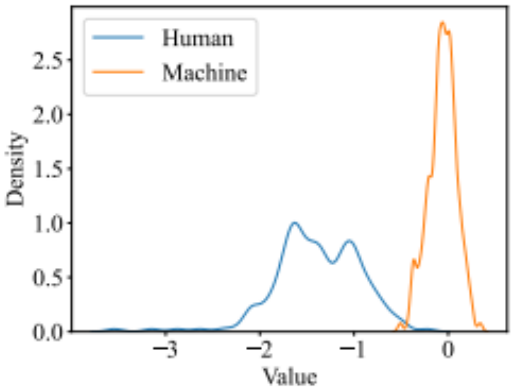


Figure 3: Likelihood gap: difference on text-davinci-003 generation on Reddit prompts [8].

Divergent N-gram Analysis (DNA-GPT) was developed for both black-box and white-box applications; however, given recent industry trends, emphasis is primarily on black-box applications. DNA-GPT is built on the empirical observation that given appropriate preceding text, LLMs tend to output highly similar text across multiple runs of generations, which is contrary to the behaviour of human-written text. Consequently, the binary classification task is one based on the 'Likelihood-Gap Hypothesis' (Figure 3), whereby humans do not choose words based on maximum likelihood like machines do [8]. In the black-box context, inputs are compared with generated outputs using n-gram similarity. Human text is predicted to have a lower similarity, compared with the machine-generated text. Models were evaluated using AUROC, True-Positive Rate (TPR) and False Positive Rate (FPR).

The performance of the model was compared against GPTZero, Open AI's Classifier and DetectGPT. The model was applied to predict to Reddit, PubMed and XSum datasets. The WMT-2016 dataset was used for predictions in German language. In the black-box context, DNA-GPT achieved an AUROC performance between 91% and 99% and a TPR between 29% and 91%. Significantly, the model achieved AUROC values between 90% and 92% on German text. Consequently, DNA-GPT achieved superior performance compared to supervised baselines GPTZero and OpenAI's Classifier across both old and new datasets, in both German and English language [8].

The performance range of DNA-GPT is comparable to trained LLMs. Significantly, the model score on German text suggests its potential for generalisability across (at least some) European Languages. Lower performance on German text was attributed to lower data volume; consequently, greater data volume may address this shortcoming. The likelihood-gap hypothesis is a strength of this model, which may make the model more robust to the growing sophistication of LLMs. This factor will be explored in more detail in section 3.

d) Watermarking methods

Yang et al. developed a watermarking method which injects BERT-generated synonyms into generated text [9]. Words representing bit-0 are selectively replaced by synonyms with a bit-1 value, thereby creating a statistically improbable sequence of bit values. Statistical testing is then employed to test for randomness, which identifies the improbable (watermarked) text (Table 1) [9]. AUROC was chosen as the appropriate evaluation metric on the HC3 dataset. This resulted in performance metrics of up-to 100%.

Although the performance metrics suggest the efficacy of black-box watermarking, the practical application of this technology to the use case is problematic. It is imagined that the end user would attempt to identify AI text generation from a previously unseen piece of text. Consequently, she does not have access to the LLM. Moreover, were there to be a detector released by a technology company based on watermarking their language model, it seems plausible that those seeking to plagiarise would avoid using such an LLM and opt to use a different one instead. Consequently, deployment of this technology to this particular use case does not seem feasible at present.

Table 1: Examples of watermark detection. Th p-value indicates the likelihood of the text not containing a watermark [9].

	Text Content	*p*-value	
		Fast	Precise
Original	Flocking is a type of coordinated group behavior that is exhibited by animals of various species, including birds, fish, and insects. It is characterized by the ability of the animals to move together in a coordinated and cohesive manner, as if they were a single entity. Flocking behavior is thought to have evolved as a way for animals to increase their chances of survival by working together as a group. For example, flocking birds may be able to locate food more efficiently or defend themselves against predators more effectively when they work together.	0.9933	0.9646
Watermarked	Flocking is a kind of coordinated team behavior that is exhibited by animals of several species, notably birds, fish, and insects. It is characterized by the ability of the animals to move together in a coordinated and cohesive way, as if they were a single entity. Flocking behavior is believe to have evolved as a way for animals to raise their likelihood of survival by working together as a group. For instance, flocking birds could be able to locate nutrition more efficiently or defend themselves against predators more effectively when they work together.	0.0342	0.00004

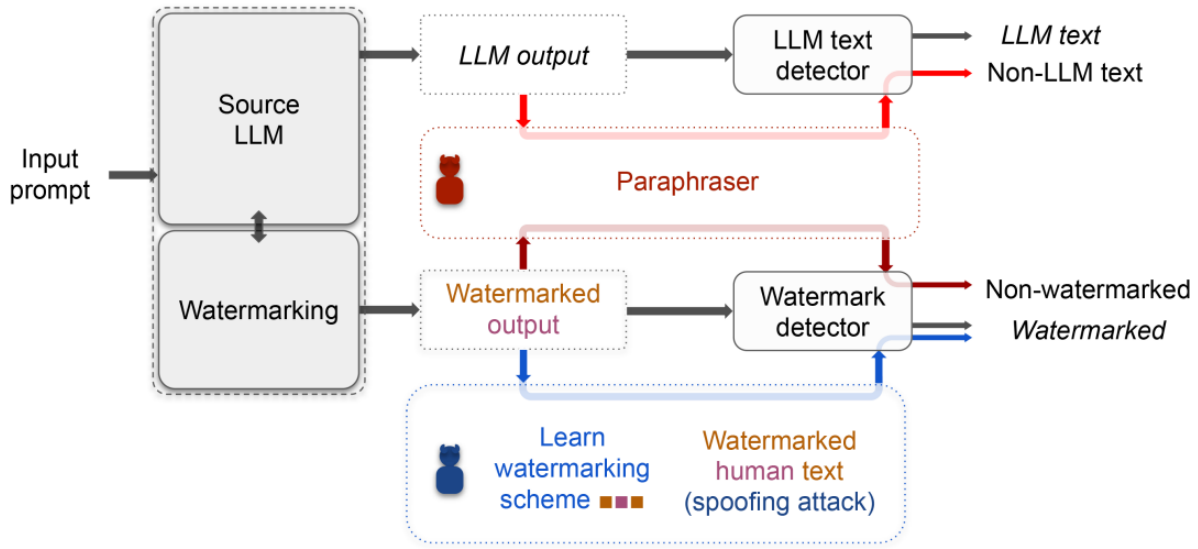


Figure 4: An illustration of the vulnerabilities described by Sadasivan et al. [10].

3) A Review of Existing Text-Identification Technology

Since it is not known which LLM a potential offender may use, generalizability is the primary challenge for developers. Considering this, together with current industry tendencies (not to release product source code), white-box approaches will not be suitable for this use case. As black-box watermarking presents similar challenges, they are also unsuitable.

Given the inapplicability of white-box solutions, the following methods remain for consideration: training methods using traditional approaches, training methods involving the tuning of LLMs and zero-shot methods. Different methods have different strengths, with traditional methods being more cost effective, LLM-based ones being more generalizable and ROBERTa-MPU excelling in cases which require detection of shorter text strings. The review has found that applicable SOTA models tend to have an AUROC greater than 90%. In deciding on the appropriate method, consideration should be given to available financial, human and technical resources, together with the results from domain-specific deployment to the use case.

C. The Problem of Text Classification

1) The Impossibility Hypothesis

Considering the increasing sophistication of LLMs, the question of whether detectors will ever be capable of reliably distinguishing machine-generated text from human-generated text has received much attention. Sadasivan et al. found that state-of-the-art (SOTA) detectors for AI-generated text are not reliable in a practical context. They test numerous detectors using light paraphrasing and find that even with a naïve approach, the performance of all detectors drops significantly with the introduction of paraphrasing (Figure 4). For example, zero-shot detectors reduced from 96.5% to 25.2% accuracy, with the introduction of paraphrasing; similarly, the TPR of Open AI's RoBERTa detector reduced from 100% to 60%. Moreover, they found that retrieval-based detectors, which are designed to evade paraphrasing attacks, are vulnerable to recursive paraphrasing. Sadasivan et al. found that with five rounds of recursive paraphrasing the detection accuracy on a retrieval detector drops to 25%.

Spoofing attacks pose another risk to the performance of detectors. Sadasivan et al. highlighted privacy issues regarding the practice of storing users' input prompts, thereby presenting a further challenge to the practicality of retrieval models [10]. This research also introduces the concept of 'spoofing attacks', whereby adversarial humans may infer the watermarking scheme through prompting, thereby enabling the deciphering and introduction of patterns to the text to mimic watermarks. Most importantly, Sadasivan et al. reached what they term a theoretical impossibility result. This impossibility result states that as LLMs become more sophisticated, they will resemble human-generated text more closely; consequently, the distribution of AI-generated texts more closely resembles that of human-generated texts. Therefore, distinguishing between distributions will become more difficult [10].

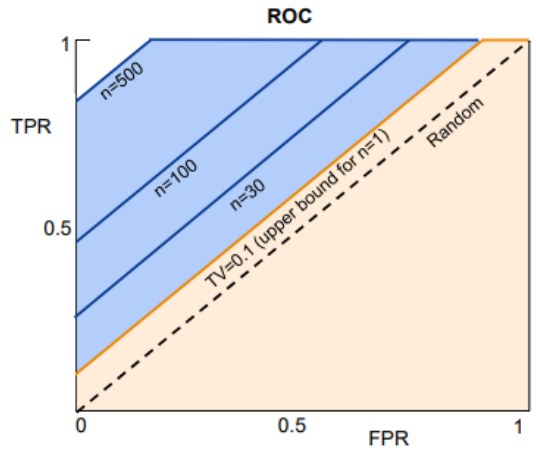
2) The Possibility Hypothesis

One significant response to the impossibility hypothesis, above, has been that of Chakraborty et al. The challenges posed by the improvements in LLMs is interpreted as a challenge of data deficiency. It is asserted that unless the human-generated and machine-generated text distributions are indistinguishable across their *entire support*, distinguishing between human- and machine-generated text is *consistently achievable*, when more data or greater sequence length is available to the detector [11]. This point is emphasized by illustrating ROC moving towards a higher T *Figure 5: Influence of number of samples on ROC curve [11].*

PR and lower FPR as the size of the sample increases (Figure 5). Subsequently, this finding is illustrated mathematically.

The generated hypothesis is then validated experimentally, using various machine learning models and SOTA detectors, all of which show improved performance with increased sequence length or data input. As was anticipated by the work of Sadasivan et al. [10], the performance of the models decreased with the introduction of

paraphrasing attacks, showing a performance reduction of 15%. However, the AUROC still increases with increases in sequence length, thereby validating the data-deficiency hypothesis [11].



Finding that it is always possible to distinguish between human- and machine-generated text accords with the likelihood gap hypothesis proposed by Yang et al., which assumes that machines maximize the log probability function (Figure 3) [8]. In contrast, humans do not generate language according to statistical probability. It seems that the greater the data volume, the more distinct these distributions will be, as more data is generated using distinct processes.

3) The Impossibility Hypothesis: An Assessment

Whilst it seems inevitable that improving LLMs will precipitate technical problems for the field, it does not follow that the probability distributions of LLM-generated text will converge with those of human-generated text, as has been asserted by Sadasivan et al. [10]. First, it has not happened to date and, secondly, previous work has shown that the means by which AI-generated language is generated is fundamentally different to that of humans [8], [11]. This difference has the potential to be leveraged by AI detectors. Therefore, in the absence of a fundamental shift in how AI generates language, the impossibility of distinguishing between human- and machine-generated text seems unlikely to manifest in the near term. Given the reality of the challenges posed by LLMs; the possibility of detection has the potential to be developed into tools which can be deployed in practical settings, thereby addressing domain concerns associated with AI-generated student essays.

Table 2: AI-text detectors accuracy score [16]

Feature Group	Descriptions	Feature Count
Basic NLP	char count, word count, word density, punctuation, title word count, upper-case count, noun count, adv count, verb count, adj count, pro count	11
Term Frequencies and NGram	Count_vect	35742
	Bigram_words	5000
	Trigram words	5000
	BiTrigram chars	5000
Topic modelling	NeuralLDA [58]	20
Others	Readability score, NER count, text error length	10

D. Application to Student Essays

1) Approaches

There are several ways in which Machine Learning (ML) can be used to detect AI-created text. Crothers, Japkowicz, and Viktor propose to divide all ML approaches into two main categories [12]:

- Feature-based [12].
- Neural Language models (NLM) [12].

The same classification is used by Tang, Ruixiang, Chuang, and Hu in their research where they refer to them as:

- Traditional Classification Algorithms [2]
- Deep Learning Approaches [2].

These two approaches work in the following ways:

- Feature-based/Traditional Classification Algorithms use various features extracted from a text within a classification algorithm such as Decision Trees, SVM, and others to classify if the text is AI-created [12][2].
- Neural Language Model/Deep Learning Approaches use finetuned large language models (LLM) such as BERT/RoBERTa to classify if the text is AI-created [12][2].

In the sections below, these two approaches will be described in more detail and for consistency we will refer to them as to Feature-Based and NLM approaches.

2) Feature-Based approach

This approach exploits the idea that AI-created text contains several features that are only specific to AI-created text. This idea is formed from observations that LLMs create different artifacts in text they generate [12]. Text features such as frequencies, linguistics, fluency, and fact verification should be considered as the most important [13].

a) Feature Extraction

One of the ways to extract features from text is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF allows the conversion of text into numerical representation [16].

TF-IDF was successfully used by Hayawi, Shahriar, and Mathew in their work. It allowed them to create AI-detection models with an accuracy of up to 95% for essays, 84% for stories, and 93% for poetry [14].

Similarly, Alamleh, AlQahtani and ElSaid used a similar approach in their work. They created models with an accuracy of up to 93% for essays, 93.5% for programming text, and 92.5% for both [16].

Trung, Hatua, and Sung in their work used more advanced feature extraction. Their feature engineering consisted not only of TD-IDF features, but also of NLP-related features, N-gram features, topic modelling features, readability scores, Named Entity Recognition (NER) counts, and text error length features (Table 1) [13]. To determine the best features, they used Principal Component Analysis (PCA) [13].

This set of features allowed them to create models that were able to predict Wikipedia-based and US Election 2024 news article-based text with impressive accuracy 99% [13].

We can assume, that TF-IDF shows good results by itself, but additional feature extractors can boost model performance even more.

b) Modelling

Looking at performance tests conducted by different research groups, it is hard to determine the most well performing ML algorithm. It seems that the algorithm performance heavily depends on the type of dataset. Overall, we can say that the most promising algorithms are Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR):

Tests performed by Alamleh, AlQahtani and ElSaid [16]:

Table 3: AI-text detectors accuracy score [16]

Model	Essay Prompts	Programming Prompts	Combination
RF	93%	93.50%	92.50%
SVM	91.50%	91%	91%
LR	92.50%	85.50%	88%
DT	87%	87.50%	90.25%
KNN	65.50%	70%	61.75%
NB	89.50%	82.50%	88%
GB	92%	91%	88.75%
FLM	91%	89.50%	91.25%
GLM	91.50%	88.50%	91%
FNN	90.50%	86.50%	91.75 %
BERT	73.46%	62.00%	69.69%

Test performed by Trung, Hatua, and Sung [13]:

Table 4: AI-text detectors accuracy score [1]

	Accuracy	Precision	Recall	F1-score
RF	0.9993	0.9992	0.9993	0.9993
SVM	0.7421	0.7422	0.7389	0.765
XGBoost	0.9993	0.9993	0.9993	0.9993

Tests performed by Hayawi, Shahriar, and Mathew [14]:

Table 5: Essay accuracy score [14]

Model	Accuracy	Precision	Recall	F1-Score
Human vs GPT				
RF	0.7536 ± 0.0308	0.84	0.63	0.63
SVM	0.7983 ± 0.0332	0.80	0.73	0.75
LR	0.6828 ± 0.0121	0.84	0.53	0.46
LSTM	0.6642 ± 0.0031	0.33	0.50	0.40
Human vs BARD				
RF	0.9812 ± 0.0138	0.98	0.98	0.98
SVM	0.9875 ± 0.0078	0.98	0.99	0.99
LR	0.9875 ± 0.0102	0.99	0.99	0.99
LSTM	0.9477 ± 0.0355	0.95	0.94	0.94

Table 6: Story classification score [14]

Model	Accuracy	Precision	Recall	F1-Score
Human vs GPT				
RF	0.9918 ± 0.0025	0.99	0.95	0.97
SVM	0.9985 ± 0.0014	1.00	0.99	0.99
LR	0.9854 ± 0.0051	0.99	0.90	0.94
LSTM	0.9940 ± 0.0052	0.99	0.97	0.98
Human vs BARD				
RF	0.9861 ± 0.0035	0.99	0.91	0.94
SVM	0.9966 ± 0.0008	1.00	0.98	0.99
LR	0.9775 ± 0.0039	0.99	0.85	0.90
LSTM	0.9914 ± 0.0025	0.99	0.95	0.97

3) NLM approach

This approach uses already created LLM such as GPT, BERT, or RoBERTa as a foundation [2]. Then they are finetuned to be able to capture distinctions between AI and human created text [2]. As a result, this can be used to detect if the text is created by AI.

The training process is similar to training processes of other classification LLMs.

a) Comparison with Feature-Based approach

The biggest advantage of NLM approach is often much superior detection outcomes [2]. In one example a pre-trained RoBERTa model trained on 8000 essays was able to have accuracy rate 99.75% versus Feature-Based approach with accuracy 95% [15].

While deep learning approaches often yield superior detection outcomes, they have disadvantage in the form of their black-box nature [15],[2]. It means that their black-box nature severely restricts interpretability and transparency [2], which is important for such tasks where evidence collection is important [15].

Model	Essay Prompts	Programming Prompts	Combination
RF	0.05 seconds	0.06 seconds	0.10 seconds
SVM	0.00 seconds	0.00 seconds	0.00 seconds
LR	0.00 seconds	0.00 seconds	0.01 seconds
DT	0.01 seconds	0.00 seconds	0.02 seconds
KNN	0.00 seconds	0.00 seconds	0.01 seconds
NB	0.00 seconds	0.00 seconds	0.00 seconds
GB	0.16 seconds	0.07 seconds	0.25 seconds
FLM	0.00 seconds	0.00 seconds	0.00 seconds
GLM	0.00 seconds	0.00 seconds	0.00 seconds
FNN	0.62 seconds	0.37 seconds	1.23 seconds
BERT	> 1000 seconds	> 1000 seconds	> 1000 seconds

Table 8: Training time for different approaches [16].

Another disadvantage is the need of NLM to have vast amount of data and computing resources for training [13]:

In Table 2 we can see the performance of an NLM model trained on 500 data entities [16]. The accuracy of the BERT model for essays is only 73.46%, while the accuracy of Feature-Based model is 93% [16]. Such low accuracy is the result of small dataset [16].

Table 7 (Appendix 1) shows amount of time needed for Feature-Based and NLM (BERT) approaches [16]. The BERT model requires more than 1000 seconds where RF model only 0.05 seconds:

4) Implication of knowledge

Looking at the information provided before, it can be suggested that a Feature-Based approach shows better performance over the NLM approach. This is further proved by the test conducted by Alamleh, AlQahtani, and ElSaid [16].

Further analysis shows that feature engineering plays a crucial role in the performance of Feature-Based models. TF-IDF by itself gives promising results, creating models with an accuracy up to 95%. While this is a good result, manual feature engineering, which uses additional algorithms, can create models with the accuracy up to 99%.

ML algorithms such as Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR) showed great results in detecting AI-created text [16] [14] [13].

Looking at that, it is assumed, that a Feature-Based approach which uses TF-IDF feature extraction and RF, SVM, or LR algorithms will enable the achievement of project deliverables. This solution potentially will allow us to create models with the accuracy up to 95%.

The accuracy of the model can be improved further with the use of manual feature engineering. This has the potential to achieve models with an accuracy of up to 99%.

To improve existing solutions, TF-IDF will be augmented with additional feature engineering to see what works the best. This is done with the aim of finding solutions with the biggest impact and the least possible complexity, which potentially will allow them to be used in already existing systems. Additionally, the possibility of utilizing NLM for feature extraction will be considered.

E. Proposed Work

Based on the above work it has been shown that AI text detectors can contribute to the issues raised by the growing use of LLMs. Towards this end, numerous methods have been developed and deployed with the aim of distinguishing machine- from human-generated text. Whilst the performance of many models is vulnerable to paraphrasing attacks, both theory and evidence posit that, in the near term, increasing the sophistication of text detectors can address the challenges posed by more powerful LLMs. Zero-shot and training methods have the capacity to produce high-performing products. This is evidenced by the application in both the domain of student essays and the general text-classification context. Domain-specific research, however, shows that the optimal solution is largely dependent on the characteristics of the available dataset, particularly the available data volume. Moreover, with the addition of automated and manual feature engineering, traditional, feature-based approaches have the capacity to yield performances on par with more sophisticated NLM-based technology. Since NLM methods require significantly more computational power to perform, end-user computational capacity should be considered when deciding on an appropriate research path. Similarly, whereas most research consulted in this review has emphasized AUROC as a performance metric, the addition of FPR-minimization as a research objective should be given serious consideration. There are numerous risks associated with false accusations of plagiarism, including liability and reputational damage –both to the student and the institution.

III. Test Design

The research objective of this project is to create a machine learning solution which can reliably distinguish between AI-generated and human-written essays. This is a supervised learning problem. The purpose of the section which follows is to provide a high-level overview of the process which aims to achieve this objective, as well as the ethical considerations in which the project is situated.

A. Ethical Considerations

The purpose of this research project is to contribute to the promotion of integrity in academia and higher education. This endeavor is likely to receive a positive evaluation by observers. However, there are numerous risks which should be kept in mind by end users and researchers. These include data management and potential harm to persons; these issues are addressed in the sections which follow.

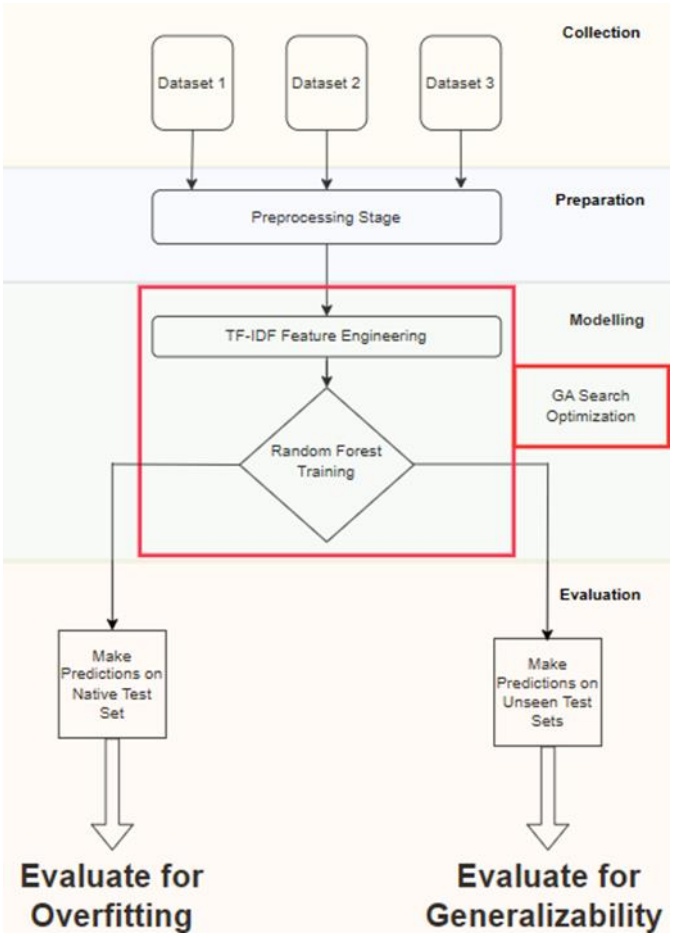


Figure 6 Design Flow Chart

1) Data Management

All data utilized in the design and training of this project were open source. Therefore, many issues which should be considered do not arise in this instance. The first of these is the recording and maintenance of student essays, together with personal information. It is recommended that appropriate consideration is given to issues of intellectual property and data protection, where non-open-source data is being utilized.

2) Harm to Persons

The potential for wrongful accusation is the most pertinent issue with the deployment of any model involving accusations of plagiarism. Such instances have the potential to cause harm to those accused, including damage to reputation, academic progress, career progression, financial wellbeing and mental health. These outcomes are also associated with liability for the accusing institutions. Therefore, it is paramount that such outcomes be protected against. Consequently, this test design includes false-positive rate as a performance metric; this can assist the end user in evaluating the suitability of the model with these risks in mind [1].

B. Data Collection

Data is collected from several datasets. Where only one dataset is used, the model may assume that all essays have the characteristics of those in the training set. Having more than one dataset enables the researcher to mitigate this risk by testing a trained model on a non-native dataset.

C. Data Preparation

1) Preprocessing stage

a) Formatting

All datasets are converted into .csv format, where they were not originally in that format.

b) Target feature creation

A binary feature is used to easily distinguish between classes in a binary classification problem. In this design, AI-generated cases are assigned a value of 1 and human-written, a value of 0. These values are obtained from the existing features in the collected data.

c) Data splitting

The data was first split into feature and target sets, with the text feature as the input and the created binary feature as the target. Then, data is split into training and test sets, using a 75/25 split. The default split was chosen, to balance the demands of validation with the computational budget.

2) TF-IDF Feature Engineering

Term frequency-inverse document frequency (TF-IDF) represents the importance of a term in a document relative to a collection of documents. The literature review found that good performance in text classification tasks could be achieved through the employment of this statistic. TF-IDF is the product of term frequency and inverse document frequency –where term frequency relates to the number of times a given term appears in a document and document frequency relates to the number of documents (in the collection) in which the term appears. The following is a mathematical expression of TF-IDF:

$$TF - IDF(t, d, D) = TF(t, d) . IDF(t, D).$$

D. Data Modelling

1) Random Forest

Random forest is suitable for supervised machine learning problems. It implements majority voting in numerous decision trees to predict the target class. The literature review found that the application of random forest in conjunction with TF-IDF has been successfully applied to text classification problems. Therefore, it was considered appropriate to apply to this research question. As above, in an exploratory context, in the absence of a baseline model, the default setting of 100 trees was deemed appropriate at this stage.

2) Genetic Algorithm for Optimization

Genetic algorithm belongs to a family of algorithms called 'Evolutionary Algorithms'. It attempts to emulate the Darwinian process of natural selection. This process results in improvements through iteration, where optimization is achieved by using fitness as the objective function [29]. Genetic algorithms have been successfully applied to random forest for hyperparameter optimization. This has yielded positive results in classification problems [30].

E. Model Evaluation

Each model is tested on both the native test set (that is, the test set from the original dataset of the training model) and unseen test sets. The following metrics were deemed suitable for model evaluation: F1 score, accuracy and (low) false-positive rate. Multiple performance metrics are used to give further insight into both the overall performance of the model and the suitability of the model for deployment in the outlined use case.

1) F1-score

F1-score is the harmonic mean of precision and recall. It considers both false positives and false negatives, and it is employed when classes in the dataset are imbalanced. The value for this metric is calculated using the following equation:

$$F1 = 2 * precision * recall / precision + recall$$

2) Accuracy

Accuracy is given by the following equation:

Accuracy = Correct Predications/Total Predictions

This metric is a suitable performance metric in binary classification tasks when datasets are balanced.

3) False-Positive Rate

Given the risks of wrongly accusing a person of cheating, achieving a low value for this performance metric is important in this use case. It is given by the following equation:

False Positive Rate = False Positives/False Positives + True Negatives.

IV. Test Configuration

A. Data Collection and Description

There were seven datasets collected in this project. The first three were used to train and optimize the model. Then, it provided the native test set for the first evaluation. The section which follows provides a brief description of the collection process and a summary description of the data.

1) Dataset 1: Hayawi et al (2023) [26]

This dataset is publicly available and was downloaded from the github page of the authors [1]. This dataset contained over 17,000 human-written essays, which were obtained from the William and Flora Hewlett Foundation. The 398 AI-generated essays were generated using BARD and ChatGPT3.5.

2) Dataset 2: Oketunji (2023) [27]

This dataset is publicly available on HuggingFace. Unlike the other datasets, this dataset is balanced, with approximately one million observations for each class, human-written and AI-generated. However, due to computational expense, only a random sample of 200,000 observations were utilized.

3) Dataset 3: Verma et al. (2023) [28]

This dataset is publicly available on the primary author's github page. The dataset is unbalanced. There are 1,000 human-written essays, which were obtained from IvyPanda. Also, there are 6,000 AI-generated essays, which were generated using various LLMs.

4) Dataset 4: King et al. (2023) [31]

This dataset is publicly available on Kaggle. The dataset is unbalanced: it contains 1,375 AI-generated essays and only 3 human-written essays. The authors do not specify which LLMs were used to generate the AI-generated essays.

5) Datasets 5 & 6: Osmulski (2023) [32]

These datasets are publicly available on Kaggle. They are imbalanced: they only contain only AI-generated essays. The 500 essays in dataset 3 were generated using GPT-3.5 turbo and the 200 essays in dataset 6 were generated using GPT-4.

6) Dataset 7: Corizzo and Leal-Arenas (2023) [33]

This dataset is publicly available on the primary author's GitHub page. The dataset is almost perfectly balanced: there are 336 AI-generated essays and 335 human-written essays. The human-written essays are taken from a sample of 1,489 essays written by Swedish university students at three different proficiency levels. All AI-generated essays were created by ChatGPT using the prompt "write an 800-word essay on [topic]", where the topics were already present in the collection of human-written essays.

B. Data Preparation

1) Formatting

All datasets were converted into .csv format. A new column was defined according to how the essay was generated. A value of 1 was assigned where the essay was AI-generated, 0 where the essay was human written. All other, non-relevant, columns were then deleted. In the cases of datasets 1 and 3, smaller datasets were then combined to form the final dataset. Finally, datasets 1, 2 and 3 were combined to create a single, larger set for model training. Later, this combined set was used to provide the native set for the evaluation phase.

2) Partitioning the dataset

All datasets were split into input and target features. The essays were used as an input and the assigned class (human-written or AI-generated) was used as a target feature. Next, the dataset was partitioned: 25% of the data was reserved for a test set; the remaining 75% was allocated to the training set.

C. Data Modelling

1) Feature engineering and feature selection

Input features were engineered from the essays using the TfidfVectoriser() function in the sklearn library. This was accomplished by creating a vectorizer by fitting the TfidfVectoriser() function on the essays in the training set. Then, the trained vectorizer was applied to both the essays in the training set and the essays in the test set. This resulted in the generation of input features. 5 hyperparameters were determined using the genetic optimization strategy in conjunction with 5-fold cross-validation. The choice of hyperparameters was informed by the literature review combined with domain knowledge, and the algorithm was trained to optimize F1-score. The chosen hyperparameters were the following: exclusion of certain words, ngram range, sublinear scaling, maximum term frequency and minimum term frequency.

2) Random Forest for classification

A Random Forest classification model was trained on the prepared training set using the RandomFotestClassifier() function in the sklearn library. The hyperparameters were determined as part of same optimization process as the feature engineering and selection phase, above. Similarly, the choice of hyperparameters for optimization was informed by a combination of the literature review and domain knowledge; they comprised the following: the number of estimators, the max depth, and the evaluation criterion. The code optimization is below (Figure 7):

```
# Create pipeline
pipe = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('rf', RandomForestClassifier())
])

# Setup crossvalidation
cv = KFold(n_splits=5, shuffle=True)

# Setup range of hyperparameters
param_grid = {
  'tfidf__stop_words': Categorical(['english', None]),
  'tfidf__ngram_range': Categorical([(1, 1), (1, 2)]),
  'tfidf__sublinear_tf': Categorical([True, False]),
  'tfidf__max_df': Continuous(0.5, 1.0),
  'tfidf__min_df': Continuous(0.001, 0.3, distribution='log-uniform'),
  'rf__n_estimators': Integer(50, 200),
  'rf__max_depth': Integer(1, 20),
  'rf__criterion': Categorical(['gini', 'entropy', 'log_loss'])
}

# Setup callback
callback = ProgressBar()

# Create estimator
evolved_estimator = GASearchCV(
  estimator=pipe,
  param_grid=param_grid,
  cv=cv,
  scoring='f1',
  population_size=10,
  generations=20,
  keep_top_k=3,
  algorithm="eaMuCommaLambda",
  n_jobs=-1
)

# Start searching optimal hyperparameters
evolved_estimator.fit(X_train, y_train, callbacks=callback)
```

Figure 7: Optimization pipeline and execution

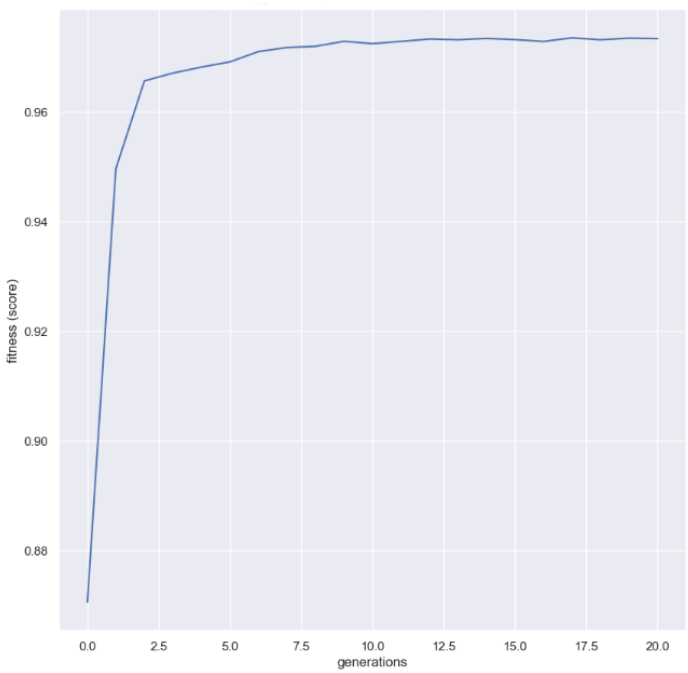


Figure 1:Fitness evolution over generations.

D. Model Evaluation

First, the model was tested on the native test set (the test portion of merged datasets 1, 2 and 3). Next, the model was tested on completely unrelated, unseen datasets (datasets 4, 5, 6 and 7). Additionally, models were tested on the full unmerged native datasets (datasets 1, 2 and 3) (Appendix 1). The model was evaluated in the context of the following performance metrics: F1-score, accuracy, and precision. Multiple performance metrics were utilized to yield more insight into the suitability of the model to the use case, given the ethical considerations.

V. Results

The model scored an F1-score of 0.997 on the training set and 0.983 on the test set. The confusion matrix is below (Table 9).

Table 9: Confusion matrix from the native test set.

	True (Predicted)	**False (Predicted)**
True (Actual)	3782	65
False (Actual)	109	6814

Table 10: Model performance on native and unseen datasets.

	F1-score	**FPR**	**Accuracy**	**Precision**
Native set (1-3)	0.98	0.016	0.98	0.97
Dataset 4	0.14	0.007	0.99	0.09
Dataset 5	0.98	NA	0.96	1.00
Dataset 6	0.98	NA	0.96	1.00
Dataset 7	1.00	0.00	1.00	1.00

VI. Discussion

A. Performance on Native Dataset

The model performed well on the native test set. Although the performance decreased slightly, from an F1-score of 0.99 on the training data, the F1-score on the test set was still 0.98, which is consistent. The retention of performance between these two tests indicates that the model is well fitted to the training data: there is no evidence of overfitting.

There were 3,847 AI-generated and 6,923 human-written essays in the test set, respectively. Therefore, the test set is slightly imbalanced. However, the dataset was of significant scale, and both classes were present in sizable quantities. In this context, the model predicted both classes with high accuracy, thereby indicating its proficiency in identifying the characteristics of each respective class. Likewise, with an FPR lower than 2%, the model is conducive to a low-risk product, even in the absence of complementary strategies. The model would result in a low quantity of false accusations.

B. Performance on Unseen Datasets

1) Dataset 4

Dataset 4 is heavily imbalanced. Whilst there are 1,465 human-written essays, there are only 3 AI-generated essays. The model performed poorest on this dataset, with an F1-score of 0.14. Of the 3 AI-generated essays, only 1 was correctly classified. Despite this, however, the model showed proficiency at correctly classifying human-written essays. For this reason, FPR was 0.007. An FPR of below 1% demonstrates that the model is at low risk of false accusations. Additionally, it shows model proficiency regarding the problem of the use-case. With the ability to correctly classify over 99% of human-written essays, the model has the potential to contribute to the mitigation of the risks posed by false accusations.

2) Dataset 5

Dataset 5 is completely unbalanced: it comprises 500 essays, all of which are generated using GPT-3.5 turbo. The model performs very well on this dataset, with an F1-score of 0.98. Similarly, the model incorrectly classifies only 3.8% of AI-generated essays. Given that the data was trained on a dataset with more human-created essays than AI-generated ones, there is no evidence that the model is more inclined to simply choose the majority class of the training set. This dataset would indicate that the model is proficient at correctly classifying the target based on the characteristics of input features. This supports the potential generalizability of the model to the use case.

3) Dataset 6

Like dataset 5, dataset 6 is perfectly unbalanced. Dataset 6 contains only 200 essays, all of which are AI-generated. Moreover, all these essays were generated on GPT-4, a much more sophisticated LLM than that used to generate many other AI-generated essays. The performance of the model on GPT-4-generated essays is similar to that of GPT-3.5 turbo-generated essays. However, somewhat unexpectedly, model performance on GPT-4-generated essays is slightly better, with a misclassification rate of only 3.5%, compared to 3.8% on GPT-3 turbo. As was the case with dataset 5, this dataset demonstrates the generalizability of the model to the use case. However, dataset 6 also demonstrates that the model is likely to be robust to the incremental improvements of LLMs, as they become more powerful.

4) Dataset 7

Dataset 7 is almost perfectly balanced. There are 335 human-written essays and 336 AI-generated essays. The model produces near-perfect results on this dataset. It correctly classified all human-written essays and only misclassified 1 AI-generated essay. As was the case with other unseen datasets, the performance of the model on this dataset demonstrates, first, the generalizability of the model and, secondly, the potential of the model for deployment to the use case.

VII. Conclusion

This project aimed to construct a binary classifier, which could accurately detect AI-generated essays. To solve this problem, a random forest model was trained on a dataset comprised of smaller datasets from numerous sources. A TF-IDF vectorizer was used to extract features from the text, and the model was optimized using genetic search algorithm, where fitness was optimized to achieve the optimal F1-score. The model was tested on the native test set; this was partitioned from the original combined dataset. There was no evidence of overfitting to the training set. Additionally, the model was tested on unrelated, unseen datasets. The strong scores on non-native sets demonstrated the generalizability of the model. There was no evidence that the model was substantially biased by the imbalance in the training set. Also, the model performed similarly, regardless of the sophistication of the generating LLM.

Considering the impressive performance on both the native and non-native test sets, the model shows promise for deployment. However, it should be noted that the model has not achieved a perfect score on the FPR. Therefore, it may not be feasible to deploy the model in its current state, and it may be required that the model undergo further development.

Further research in this area should include investigating the feasibility of optimization based on FPR. Ideally, this would be reduced to 0. Secondly, similar models should be constructed using alternative machine learning models, such as support vector machines or logistic regression. These may yield results which improve upon the benchmark established in this project. Thirdly, further attention should be paid to essays constructed using more powerful LLMs; this will help establish the extent of robustness to more sophisticated LLMs. Although the model was found to be robust against GPT-4, it was only tested on a set of 200 essays from this LLM. Testing on a larger set of such data is required before reaching firm conclusions on model robustness to GPT-4. Finally, whilst such a large computational budget was not assigned to this project, the model may be improved upon by training on larger, more varied datasets. This is a step which should be performed prior to deploying any product to market.

REFERENCES

[1] X. Yang *et al.*, 'A Survey on Detection of LLMs-Generated Content', University of Califrnia, Santa Barbara, Santa Barbara, California, Oct. 2023. Accessed: Feb. 02, 2024. [Online]. Available: https://arxiv.org/abs/2310.15654

[2] R. Tang, Y.-N. Chuang, and X. Hu, 'The Science of Detecting LLM-Generated Texts', Department of Computer Science, Rice University, Houston, Texas, Jun. 2023. Accessed: Feb. 02, 2024. [Online]. Available: https://arxiv.org/abs/2303.07205

[3] S. Gehrmann, H. Strobelt, and A. M. Rush, 'GLTR: Statistical Detection and Visualization of Generated Text', Harvard University, Jun. 2019. Accessed: Feb. 02, 2024. [Online]. Available: http://arxiv.org/abs/1906.04043

[4] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, 'DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature', Jul. 2023.

[5] P. Fernandez, A. Chaffin, K. Tit, V. Chappelier, and T. Furon, 'Three Bricks to Consolidate Watermarks for Large Language Models', Université de Rennes, Rennes, Nov. 2023. Accessed: Feb. 02, 2024. [Online]. Available: http://arxiv.org/abs/2308.00113

[6] Y. Tian *et al.*, 'Multiscale Positive-Unlabeled Detection of AI-Generated Texts', Peking University, Peking, Sep. 2023. [Online]. Available: http://arxiv.org/abs/2305.18149

[7] L. Fröhling and A. Zubiaga, 'Feature-based detection of automated language models: tackling GPT-2, GPT-3', *Peer J Computer Science*, vol. 7, no. e443, p. 23, Apr. 2021.

[8] X. Yang, W. Cheng, Y. Wu, L. Petzold, W. Y. Wang, and H. Chen, 'DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text', University of Califrnia, Santa Barbara, California, Oct. 2023. Accessed: Feb. 02, 2024. [Online]. Available: http://arxiv.org/abs/2305.17359

[9] X. Yang *et al.*, 'Watermarking Text Generated by Black-Box Language Models', University of Science and Technology of China, Heifei. Accessed: Feb. 02, 2024. [Online]. Available: http://arxiv.org/abs/2305.08883

[10] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, 'Can AI-Generated Text be Reliably Detected?', University of Maryland, Maryland, Jun. 2023. Accessed: Feb. 02, 2022. [Online]. Available: http://arxiv.org/abs/2303.11156

[11] S. Chakraborty, A. S. Bedi, S. Zhu, B. An, D. Manocha, and F. Huang, 'On the Possibilities of AI-Generated Text Detection', Oct. 2023. Accessed: Feb. 02, 2024. [Online]. Available: http://arxiv.org/abs/2304.04736

[12] E. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-generated text: A comprehensive survey of threat models and detection methods," *IEEE Access*, 2023.

[13] T. N. Trung, A. Hatua, and A. H. Sung, "How to Detect AI-Generated Texts?", *IEEE Xplore*, 2024.

[14] Hayawi, Kadhim, Sakib Shahriar, and Sujith Samuel Mathew. "The imitation game: Detecting human and ai-generated texts in the era of large language models." *arXiv preprint arXiv:2307.12166*, 2023.

[15] Duanli Yan, Michael Fauss, Jiangang Hao, Wenju Cui, "Detection of AI-generated Essays in Writing Assessments", *Educational Testing Service*, 2023.

[16] H. Alamleh, A. A. S. AlQahtani and A. ElSaid, "Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning," *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA, USA, 2023, pp. 154-158, doi: 10.1109/SIEDS58326.2023.10137767, 2023.

[17] S. A. Bin-Nashwan, M. Sadallah, and M. Bouteraa, 'Use of ChatGPT in academia: Academic integrity hangs in the balance', *Technology in Society*, vol. 75, p. 11, Nov. 2023.

[18] C. Kwan Lo, 'What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature', *Education Sciences*, vol. 13, no. 4, p. 410, Apr. 2023.

[19] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, 'Chatting and cheating: Ensuring academic integrity in the era of ChatGPT', *Innovations in Education and Teaching International*, vol. Mar2023, pp. 1–12, Mar. 2023, doi: 10.1080/14703297.2023.2190148.

[20] A. Vaswani *et al.*, 'Attention Is All You Need', Google Brain, Jun. 2017. [Online]. Available: edsarx.1706.03762.

[21] T. B. Brown *et al.*, 'Language Models are Few-Shot Learners', Open AI, Jun. 2020. Accessed: Feb. 11, 2024. [Online]. Available: http://arxiv.org/abs/2005.14165

[22] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, 'Is ChatGPT a General-Purpose Natural Language Processing Task Solver?', Nanyang Technological University, Nanyang, Nov. 2023. Accessed: Feb. 11, 2024. [Online]. Available: http://arxiv.org/abs/2302.06476

[23] T. Weitzman, 'GPT-4 Released: What It Means For The Future Of Your Business', *Forbes*, Mar. 28, 2023. Accessed: Feb. 11, 2024. [Online]. Available: https://www.forbes.com/sites/forbesbusinesscouncil/2023/03/28/gpt-4-released-what-it-means-for-the-future-of-your-business/

[24] Open AI, 'GPT-4 Technical Report', Dec. 2023. Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[25] M. Schreiner, 'GPT-4 architecture, datasets, costs and more leaked', The Decoder. Accessed: Feb. 11, 2024. [Online]. Available: https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/

[26] K. Hayawi, S. Shahriar, and S. S. Mathew, 'The Imitation Game: Detecting Human and AI-Generated Texts in the Era of ChatGPT and BARD'. Nov. 12, 2023. Accessed: Feb. 25, 2024. [Online]. Available: https://github.com/sakibsh/LLM

[27] A. F. Oketunji, 'Evaluating the Efficacy of Hybrid Deep Learning Models in Distinguishing AI-Generated Text'. Huggingface, https://huggingface.co/datasets/dmitva/human_ai_generated_text/viewer/default/train?p=1&row=199, Jan. 13, 2024. [Online]. Available: https://arxiv.org/abs/2311.15565

[28] V. Verma, E. Fleisig, and D. Klein, 'Ghostbuster: Detecting Text Ghostwritten by Large Language Models'. github, https://github.com/vivek3141/ghostbuster-data. Accessed: Feb. 25, 2024. [Online]. Available: https://arxiv.org/abs/2305.15047

[29] . Katoch, S. S. Chauhan, and V. Kumar, 'A review on genetic algorithm: past present, and future', *Multimedia Tools and Applications: An International Journal*, vol. 80, no. 5, pp. 8091–8126, Oct. 2020.

[30] E. Elyan and M. M. Gaber, 'A genetic algorithm approach to optimising random forests applied to class engineered data', *Information Sciences*, vol. 384, pp. 220–234, Apr. 2017.

[31] [3] J. King, P. Baffour, S. Crossley, R. Holbrook, and M. Demkin, 'LLM - Detect AI Generated Text.' Kaggle, https://kaggle.com/competitions/llm-detect-ai-generated-text, 2023. Accessed: Apr. 09, 2024. [Online]. Available: https://kaggle.com/competitions/llm-detect-ai-generated-text

[32] [4] R. Osmuslki, 'LLM Generated Essays for the Detect AI Comp!' Kaggle, https://www.kaggle.com/datasets/radek1/llm-generated-essays/data. Accessed: Apr. 09, 2024. [Online]. Available: https://www.kaggle.com/datasets/radek1/llm-generated-essays/data

[33] [5] R. Corizzo and S. Leal-Arenas, 'One-Class Learning for AI-Generated Essay Detection'. doi: https://doi.org/10.3390/app13137901.

VIII. Appendix 1: Literature Review Table 7

Table 7: Poetry classification score [14]

Human vs GPT				
RF	0.9675 ± 0.0009	0.98	0.53	0.55
SVM	0.9941 ± 0.0014	0.99	0.92	0.95
LR	0.9752 ± 0.0007	0.99	0.65	0.72
LSTM	0.9912 ± 0.0032	0.96	0.90	0.93
Human vs BARD				
RF	0.9672 ± 0.0014	0.98	0.53	0.54
SVM	0.9931 ± 0.0026	0.99	0.90	0.94
LR	0.9764 ± 0.0019	0.99	0.66	0.74
LSTM	0.9885 ± 0.0091	0.92	0.90	0.91

IX. Appendix 2: Full List of Results

1) Dataset 1

Table 11: Confusion matrix depicting the results of model on its own test set

	True (Predicted)	**False (Predicted)**
True (Actual)	356	42
False (Actual)	38	17642

Table 12: Performance metrics of model when applied to the test set of dataset 1.

Performance Metric	**Score**
Accuracy	1.0
Precision	0.9
Recall	0.89
F1-score	0.9

2) Dataset 2

Table 13: Confusion matrix depicting the results of model on the test set of dataset 2.

	True (Predicted)	**False (Predicted)**
True (Actual)	8953	37
False (Actual)	0	9010

Table 14: Performance metrics of model 1 when applied to the test set of dataset 2.

Performance Metric	**Score**
Accuracy	1.0
Precision	1.0
Recall	1.0
F1-score	1.0

3) Dataset 3

Table 15: Confusion matrix depicting the results of model 1 when applied to the test set of dataset 3.

	True (Predicted)	**False (Predicted)**
True (Actual)	5967	33
False (Actual)	110	890

Table 16: Performance metrics of model 1 when applied to the test set of dataset 3.

Performance Metric	**Score**
Accuracy	0.98
Precision	0.98
Recall	0.99
F1-score	0.99