Detecting and Unmasking AI-Generated Texts through Explainable Artificial Intelligence using Stylistic Features

Aditya Shah*, Prateek Ranka†, Urmi Dedhia‡, Shruti Prasad§, Siddhi Muni¶ and Kiran Bhowmick‖
Department of Computer Engineering, Dwarkadas J. College of Engineering
Mumbai, India

***Abstract*—In recent years, Artificial Intelligence (AI) has significantly transformed various aspects of human activities, including text composition. The advancements in AI technology have enabled computers to generate text that closely mimics human writing which is raising concerns about misinformation, identity theft, and security vulnerabilities. To address these challenges, understanding the underlying patterns of AI-generated text is essential. This research focuses on uncovering these patterns to establish ethical guidelines for distinguishing between AI-generated and human-generated text. This research contributes to the ongoing discourse on AI-generated content by elucidating methodologies for distinguishing between human and machine-generated text. The research delves into parameters such as syllable count, word length, sentence structure, functional word usage, and punctuation ratios to detect AI-generated text. Furthermore, the research integrates Explainable AI (xAI) techniques—LIME and SHAP—to enhance the interpretability of machine learning model predictions. The model demonstrated excellent efficacy, showing an accuracy of 93%.Leveraging xAI techniques, further uncovering that pivotal attributes such as Herdan's C, MaaS, and Simpson's Index played a dominant role in the classification process.**

Keywords*—Detecting AI generated text; computer generated text; AI generated text; text classification; machine learning; pattern recognition; Stylistic features; Explainable AI; Lime; Shap

I. Introduction

Artificial Intelligence (AI) has had a significant impact on how humans perform daily tasks [1], such as composing text, in recent years. The technology behind it has improved to the point that computers are now capable of generating text that closely resembles human writing. This has resulted in issues such as circulating false information and stealing identities. It's also made things less apparent, which could be dangerous for security. Given the importance of these dangers and issues [2], it is critical that the underlying patterns used by various text generation techniques are uncovered. The research paper sets ethical guidelines for emulating human styles or perspectives by distinguishing AI-generated writing from human-generated language or examining the patterns formed by AI.

Researchers have tried several methods to understand how AI generates material. Curvature-based hypothesis and perturbation discrepancy detection of machine-generated text. The hypothesis argues that machine-generated text will be at the negative curvature and human-generated text will be at the positive curvature.If the perturbation discrepancy is more than 0, the text is machine-generated; if it goes to 0, it is human-generated [3].

As input, many textual properties such as length, punctuation, and word choice are utilized. On five models, an ensemble technique with Logistic Regression is used for binary classification (text is either human or machine-generated). Three models are utilized directly without cross-validation for multiclass classification (to determine which deep neural model was used for text synthesis) [4].

The primary focus for detecting AI-generated text is linguistic analysis [5], which breaks out syntactic patterns, word choices, and sentence structures. When a person uses too many words, repeats the same thing, or breaks the rules, this is a red signal. Investigating AI prompts and replies that don't match is crucial. If the AI model doesn't make sense or changes style, a machine may be implicated. Metadata is another option. AI creation may be indicated by unusual timestamps or IP addresses. Anomaly detection methods point out when language patterns are broken. Machine learning models trained to spot anomalies can distinguish AI writing from human-written language. Determining if writing was created by AI is complicated and ever-changing. Linguistic signals, inconsistency analysis, information inspection, stylometric quirks, bias identification, outliers, and purpose-built models are crucial.

Detecting AI-generated text remains an evolving effort, with several uncharted areas that demand attention for more robust and accurate identification. Firstly, there's a need to collect a diverse and thorough corpus of training data, spanning various AI models, linguistic styles, and genres, to ensure the detection system's adaptability. Fine-tuning detection models for specific AI language generators could improve precision by honing in on the unique attributes of each model. Contextual understanding remains a problem, as AI-generated text often lacks coherence. Developing methods that examine contextual disparities and irregularities could support the detection of AI-generated text. The rise of multimodal AI-generated content demands the development of detection models that can study and correlate text, images, and videos, expanding the scope of accurate identification. To counter evolving AI models, adversarial approaches must be adaptive, having a constant back-and-forth development between detection and generation. Ensuring real-time detection capabilities is crucial, especially for online platforms, necessitating the creation of lightweight, quick-response systems that analyze text as it's created.

This research paper aims to explore various methods for identifying AI-generated text. The paper discusses various factors that need to be considered while detecting AI-generated text. These include parameters such as average syllable count, average word length, average sentence length by word, count of functional words, punctuation count ratio, and many more. Further, it implements xAI techniques which are LIME and SHAP to assist in interpreting and comprehending the predictions provided by the machine learning models.It contributes to the continuing discussion concerning AI-generated material by throwing light on the methodologies and approaches used to distinguish between human and machine-generated text.

Section II covers a wide range of techniques for detecting and understanding AI-generated text. The section provides insights into various approaches used to differentiate between machine and human-generated content, underscoring the evolving nature of this research. Section III provides a comprehensive overview of the technologies employed in this research and Section IV discusses the proposed model. In Section V, experimentation done using fine-tuning of hyperparameters is discussed and Section VI discusses the results with the conclusion in Section VII followed by future scope in Section VIII.

II. Review of Literature

Various features extracted from the text, such as length, punctuation, and word choice, are used as input. For binary classification (text is whether human or machine generated), an ensemble technique with Logistic Regression is applied to five models().For multiclass classification (to determine which deep neural model was used for text generation), three models are used directly without cross-validation [4].

The paper explores various detection methods, including classifiers trained from scratch, zero-shot classifiers utilizing pre-trained TGMs, and fine-tuning pre-trained languages models like RoBERTa and GROVER. While the RoBERTa detector shows promising results, it requires a substantial number of training examples, making it less practical (). The paper highlights the difficulties faced by the state-of-the-art RoBERTa detector, including identifying short and fluent MGT instances, factual errors, spurious entities, contradictions, and violations of common sense reasoning [6].

In the context of text recognition using the GLTR model, the underlying assumption of their methods is to generate natural-looking text. Most systems sample from the head of the distribution through max sampling, k-max sampling, beam search,temperature-modulated sampling, or even implicitly with rule-based templated approaches. [7].

A curvature-based criterion that makes use of hypothesis and perturbation discrepancy to detect machine-generated text. The hypothesis states that if the text is machine-generated, then it will lie at the negative curvature and if it is human-generated, then it will occupy the positive curvature. If the perturbation discrepancy is greater than 0, it implies that the text is machine-generated and if the perturbation discrepancy tends to 0, it implies that the text is human-generated [3].

The author in [8] explores various approaches such as Multimodal Explanation, Deep Visual Explanation, and Deep Tensor Networks to create models that can provide explanations for their decisions using visual and textual modalities. In the context of understanding and interpreting AI models and their decisions, the paper emphasizes the crucial role of xAI. It emphasizes the need for AI systems to provide explanations for their decisions in sensitive areas like healthcare. It also presents different approaches for the explainability of AI models. The paper concludes by discussing the importance of xAI.

Researchers collected a dataset of 500 data points by gathering responses from computer science students for essay and programming assignments [9]. Each response was labeled as either Human-written or machine-generated. To analyze the text, they used a technique called Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction, which converts the text into numerical representations that machine learning models can understand.

The researchers created an open dataset for the Russian language consisting of long texts generated by different models with varying parameters and sampling methods, balanced with human-written text, and experiments with data mixing which shows that blending samples from different generative models improved the generalization ability of the detector models particularly helping RoBERTa-based models in detecting machine-generated text [10]. They further increased the input length sequence which then improved the model's understanding of the context and led to better detection performance and Multi-Task learning where the model simultaneously trains on multiple tasks, proved beneficial for improving the quality of the discriminator.

This research work focused on improving the stability of the LIME algorithm in xAI. LIME (Local Interpretable Model-Agnostic Explanations) is used to explain AI algorithms [11]. The researchers identified two main stability issues with LIME which are Segment Ordering and Region Flipping and to improve LIME's stability, they proposed two strategies: High Sample size and Average Segment Weights.

The paper [12] provides a guided tutorial of the xAI implementation in the field of Software Engineering. It provides an introduction to xAI. Further, it provides fundamental knowledge of defect prediction models. It addresses three successful case studies where xAI is used in defect prediction models.

The authors explore challenges in distinguishing Large Language Models (LLMs) and human-generated text. They derive complexity bound for detecting AI-generated text, indicating a number of samples needed for detection [13]. The researchers also discuss different existing approaches for detecting AI-generated text, highlighting the ethical concerns related to the misuse of LLMs.

In the study [14], various methods for detecting AI-generated text are explored. There are several techniques that include analyzing word pair frequency, linguistic characteristics, lexicographical features, and many more. The paper provides details on the methodology and results of each method. Further, it concludes that there is no single best method, and further evaluation of standardized datasets is necessary.

In the research, three decoding strategies are examined. They show that the improvement in these methods is for fooling humans, rather than difficult detectable text. These decoding strategies include top-k and untruncated random sampling [15]. The authors emphasize the importance of using both human and automatic detection methods to assess the humanness of text generation systems. They call for further research in improving language models, building better automatic detectors, and developing tools to improve human detection of machine-generated text.

The research proposes a classification model for detecting essays generated by ChatGPT [16]. The model is based on XGBoost. It is trained and further evaluated on a dataset generated by ChatGPT and written by humans. It also explores feature engineering for better results. It specifically explores TF-IDF and other hand-crafted features.

The research examines various Machine Translation methods and assesses the linguistic complexity of their translations in terms of both vocabulary and grammar [17]. The study uses different metrics to measure diversity, such as lexical richness and morphological variety and applies these metrics to translations produced by MT models.

This study evaluates 13 lexical diversity metrics for tracking the progression of French learners' written productions. [18]They used a semi-longitudinal corpus of learners' essays and applied random forests to predict the production wave. The metrics show varying correlations and the ability to detect differences between productions achieving 69% accuracy in predicting production waves

The study presents a variety of techniques, including linguistic analysis, frequency counting, perplexity-based filtering, and more. [14]These methods leverage different aspects of the text, such as syntactic patterns, linguistic features, and statistical properties, to differentiate between human-written and machine-generated content.

In order to analyze complex machine learning models, the study introduces a unifying framework termed SHAP. The framework determines the significance of each feature in a model for a certain prediction [19]. The framework introduces new ways and unites six current methods to enhance computational efficiency and compatibility with human intuition. To illustrate how well SHAP works at understanding model predictions, the paper includes theoretical findings, computational experiments, and user studies.

The authors of this scientific study suggest a categorization method for categorising research paper abstracts using several machine-learning approaches. The goal is to automatically classify the papers into three categories: business, social science, and science [20]. Four machine learning techniques are tested by the authors: Support Vector Machines (SVM), Naive Bayes, K-Nearest Neighbour (KNN), and Decision Tree. Tokenization, stemming, and stop word removal are used in the pre-processing of the abstracts. For feature extraction, Bag of Words and TF-IDF vectorization techniques are applied. The authors contend that the algorithm could function even better with more data. Overall, the study shows that machine learning approaches may successfully categorize research articles based on their abstracts.

The article proposes a GPT language model and investigates its Python code-writing capabilities [21]. In contrast to GPT-3's performance of 0% and GPT-J's performance of 11.4%, the researchers are able to provide better results. Furthermore, they find that frequent sampling from the model is an extremely effective technique for coming up with workable solutions to difficult problems. The model has a number of faults, including problems with binding operations to variables and docstrings that provide detailed information. Finally, the paper goes over the wider effects that utilizing strong code generation methods might have on safety, security, and economics.

In summary, the literature survey illuminates the complexities and multifaceted nature of AI-generated text analysis, revealing that advancements in the field are often accompanied by intricate challenges and unexplored areas. This research paper endeavors to offer a comprehensive solution to the intricate challenge of detecting text that originates from artificial intelligence (AI) systems.

III. Overview of Technologies Used

Technologies used in this research can be broadly classified into three groups:

A. Machine Learning Algorithms

For AI-generated text detection, various machine learning models were chosen such as Decision Tree, Random Forest, Logistic Regression, and Support Vector Machine (SVM), Gradient Boosting. These models were chosen with the specific intention of utilizing xAI.

The selection of these models for xAI was driven by several key considerations such as:

1) Interpretability: For eg, Decision Trees provide a clear and intuitive decision-making structure represented by tree-like diagrams making it easier to understand. Interpretable coefficients provided by models like Logistic Regression and SVM indicate the impact of each feature on the outcome.
2) Balancing complexity: These models strike a balance between performance and complexity. While more complex models like neural networks may help in achieving higher accuracy, they are difficult to interpret. The chosen models provide a reasonable trade-off between predictive power and interpretability.

(A) Logistic Regression [22]: It is a statistical method used for binary classification tasks by making use of the logistic function also known as sigmoid function, which transforms a linear combination of predictor variables into a value between 0 and 1. The logistic regression is expressed by:

$$P(Y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+...+\beta_pX_p)}} \tag{1}$$

(B) Decision Tree [23]: It employees the Gini index to make informed decisions during the process of creating a tree like structure for classification tasks. It uses a recursive process to partition the feature space into regions to minimise impurity and improve classification accuracy. The Gini index is a measure of impurity and its formula is given by:

$$Gini(t) = 1 - \sum_{i=1}^{C} (p(i|t))^2 \tag{2}$$

(C) Gradient Boosting [24]: It is an advanced machine learning method that builds a strong predictive model by combining multiple weak learners. It makes use of Gradient Descent optimization to minimise the predictive errors.

$$F(x) = \sum_{m=1}^{M} \gamma \times f_m(x_i) \tag{3}$$

(D) Support Vector Machines (SVM) [25]: It is a powerful classification technique that aims to find a hyperplane in a high-dimensional feature space that best separates different classes of data points. The main idea behind SVM is to maximise the margin between the classes, which is the distance between the hyperplane and the nearest data points of each class.

$$h(x) = sign(\sum_{i=1}^{n_{SV}} a_i y_i \times K(x_1 x_i) + b) \tag{4}$$

(E) Random Forest [26]: It is an ensemble learning method that combines multiple decision trees to improve the classification accuracy. It uses random subsets of data and features to build diverse trees , thus making independent predictions.

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{5}$$

B. xAI Libraries

xAI for classification refers to the application of techniques that provide transparent and interpretable explanations for the predictions made by classification models. It also refers to the concepts and techniques used to make artificial intelligence models more transparent and interpretable to humans. In classification tasks, where the model assigns input data to specific categories or classes, xAI techniques focus on revealing the contributing factors that led to a particular classification outcome. Commonly used techniques and models include LIME, feature importance, PDP(Partial Dependency Plots), and SHAP.

1) LIME - It is a technique designed to explain the predictions made by complex ML models, particularly “black-box” models, in a more understandable and interpretable way. LIME helps bridge the gap between the often opaque nature of advanced ML algorithms and the need for human-understandable explanations.
2) SHAP - SHAP, which stands for “SHapley Additive exPlanations,” is a powerful technique used in machine learning to explain the predictions of various models. It provides a unified framework for explaining the output of any machine learning model by assigning “importance” values to each feature in a prediction. SHAP values are based on cooperative game theory and offer a comprehensive understanding of feature contributions to individual predictions.

C. Stylistic features

TABLE I. STYLISTIC FEATURES AND VARIOUS SCORES CALCULATED FOR THE DATA POINTS

Linguistic Features	Scores	Description
Lexical Features	Average Word Length [27]	This gives us the average word length of the concerned text in terms of the number of characters used.
	Average Sentence Length By Word [28]	This gives us the average number of syllables used per word in the concerned text.
	Functional Words Count [29]	Functional words are grammatical connectors or mood-defining words within phrases, lacking significant linguistic value on their own.
	Punctuation Count	This gives us the ratio of the number of punctuations used to the number of characters used in the concerned text.
Readability Score	Flesch Reading Ease [30]	This metric assesses a text’s readability by analyzing its ease of comprehension. Score Range: The scores range from 0 to 100. Text with a higher score is more likely to be easier to read.
	Flesch-Kincaid Grade Level [31]	This metric estimates the U.S. school grade level required to grasp the material. Score Range: Scores can be any value greater than zero. Lower scores suggest that comprehension is possible at lower grade levels.
	Gunning Fog Index [32]	The Gunning Fog Index determines how many years of official schooling are required for a person to fully understand a text. Score Range: The scores range from 0 to 20. Lower scores indicate simpler text.
	Dale-Chall Readability Formula [33]	Dale-Chall The readability formula considers a set of well-known words and uses their presence to determine the readability of the text. Score Range: Approximately similar to US grade levels. Lower scores imply that the text is easier to read.

The research uses various stylistic features which are calculated for every text in the data set. These features include lexical features, readability, and diversity and richness of vocabulary. These features are important and used further in the research for model training and to discover patterns and information regarding the text that are not visible and perceptible to the human eye. Table I and Table II explain the various features that are noted and calculated for the texts present in the data set.

IV. PROPOSED METHODOLOGY

The proposed methodology for this research consists of two major pipelines:

1) Dataset Generation Pipeline
2) Model Training and xAI Pipeline

All the steps occurring in both pipelines are explained in detail below.

A. Dataset generation pipeline

Fig. 1 represents the flow for the creation of the two datasets used in this research.

1) Prompt generation for every data point: The proposed model’s datasets are constructed using introductions from Wikipedia articles, specifically human-generated texts.

TABLE II. Stylistic features and various scores calculated for the data points (Diversity and Richness of Vocabulary)

Linguistic Features	Scores	Description
Diversity and Richness of Vocabulary	Yule's Characteristic K [34]	Yule's Characteristic K measures text ”disorderliness” by analyzing word frequency distribution, calculating the ratio of total words to the square root of its inverse. A lower value indicates greater vocabulary diversity, while a higher value suggests more word repetition and lower vocabulary richness.
	Herdan's C [35]	Herdan's C quantifies word frequency distribution in a text by subtracting the logarithm of total words from the logarithm of unique words. It offers insights into the text's vocabulary distribution.
	Maas [36]	Maas is a measure derived using a formula by Mueller involving variables like ”logeV0,” representing vocabulary expansion, where natural logarithm is employed, and incorporating variables a, logV0, and V to indicate proportional vocabulary expansion across the text.
	Mean segmental TTR(Type Token Ratio) (MSTTR) [37]	Mean Segmental TTR (MSTTR) calculates the average Type-Token Ratio (TTR) over consecutive text segments, where TTR is the ratio of unique words to total words in a segment. It detects shifts in vocabulary diversity within the text.
	Simpson's Index [38]	Measure that quantifies the likelihood of two words randomly selected from a text being identical. The scale spans from 0, representing a state of high diversity, to 1, indicating a state of low diversity.

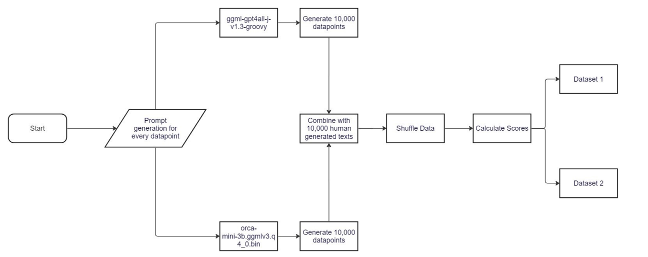
Start
Prompt generation for every datapoint
ggml-gpt4all-j-v1.3-groovy
Generate 10,000 datapoints
orca-mini-3b.ggmlv3.q4_0.bin
Generate 10,000 datapoints
Combine with 10,000 human generated texts
Shuffle Data
Calculate Scores
Dataset 1
Dataset 2

Fig. 1. Dataset creation pipeline

The creation process involves employing a prompt in the format “200-word Wikipedia-style introduction on 'title' starter_text.” Here, 'title' represents the Wikipedia page title, and 'starter_text' comprises the first seven words from the introduction paragraph of the respective article.

2) LLMs used to generate text: Two large language models, namely “ggml-gpt4all-j-v1.3-groovy” [39]and “orca-mini-3b.ggmlv3.q4_0.bin” [39], are utilized to generate a set of 10,000 data points for each model. These data points are then combined with the human-generated text, resulting in two separate datasets, each containing 20,000 data points. The final datasets are created by shuffling the data points independently for each of the models along with the human-generated text.
3) AI text generation: Both LLMs then produce 10000 AI-generated texts each for a better variety and spread of data.
4) Combining the data: All the AI-generated texts from both LLMs are then combined with 10,000 human-generated texts individually from the two LLMs.
5) Shuffling data: All of these data points are then shuffled so that the machine learning models used ahead do not learn any unintended patterns from the data, thereby impeding the performance of the model. This step denotes the creation of an intermediate datasets for this research.
6) Calculating Scores & Final Datasets: Every data point in the intermediate datasets, undergoes a series of calculations that help determine various style characteristics and linguistic features of the text such as readability, richness and correctness of vocabulary, and semantic spread of the text, and lexical features that are not lucid and discernible to the human eye. Each of these characteristics has been attributed to various scores that help determine such features in the text. These scores are then appended to the intermediate dataset thus completing the dataset generation process and thereby creating two datasets for the LLMs used.

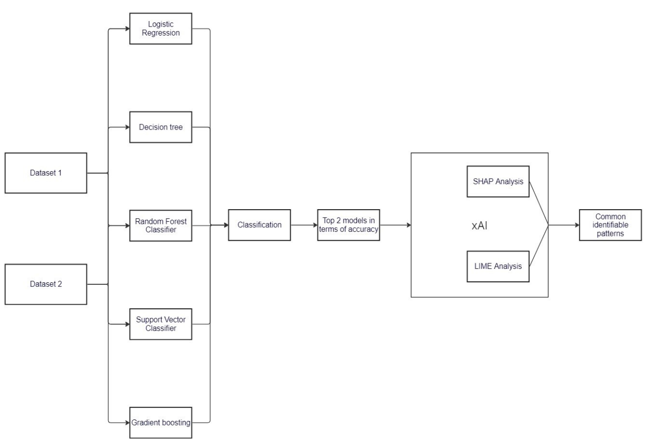

Fig. 2. Model Training and xAI pipeline

Following is the brief of all the scores used to extract the stylistic features later on in this research.

(A) Lexical Features: One of the ways in which AI generated text and human text can be distinctly identified is by its lexical structure. Usually, human text is erroneous in terms of appropriate punctuation - punctuation marks are fairly missed or used improperly. Similarly, there are disparities in other areas of lexical architectures such as the differences in word lengths and thus the number of syllables (AI tends to use heavier words), differences in typical sentence lengths, the varied usage of functional words such as she, these, or, and, etc. and so on. Hence, these factors are calculated for both classes of texts and used in this research paper to identify the trends for the same.

(B) Readability Scores: Readability scores seek to identify the reading level of a particular text, usually in terms of the minimum education level required to read the text with ease. Since human texts and AI texts are bound to differ in terms of readability ease, this paper made use of the following readability metric formulae to quantitatively identify the reading ease of human and AI generated text both:

a) Flesch Reading Ease:

$$206.835-1.015(\frac{totalwords}{totalsentences})-84.6(\frac{totalsyllables}{totalwords}) \tag{6}$$

b) Flesch-Kincaid Grade Level:

$$0.39(\frac{totalwords}{totalsentences})+11.8(\frac{totalsyllables}{totalwords})-15.59 \tag{7}$$

c) Gunning Fog Index:

$$0.4\cdot[(\frac{totalwords}{totalsentences})+100(\frac{complexwords}{totalwords})] \tag{8}$$

d) Dale-Chall Readability Formula: The Dale-Chall Formula compares its wordlist to the provided text and then determines the U.S. grade level based on the number of difficult words and average sentence length.

(C) Diversity and richness of the vocabulary: The vocabulary used by AI to generate texts and that used by human writers is vastly different. Human text considers several other factors apart from the meaning and semantics while selecting a word, such as cultural relevance, formal/informal style, text's context and usage, etc. AI on the other hand tends to focus more on the textbook definition rather than these factors, thus causing a disparity with human text even between sentences meant to convey the same meaning. The following vocabulary richness and diversity metric formulae were used in determining the vocabulary levels for both the classes of texts:

a) Yule's Characteristic K:

$$K=\frac{10^4}{N^2}\sum_{i=1}^{V}(n_i-c)^2 \tag{9}$$

Where,
N is the total number of words in the text.
V is the vocabulary size (the number of distinct terms/words).
n_i is the frequency of the ith word.
c is the mean frequency of all words.

b) Herdan's C:

$$C=\frac{\log(V)}{\log(N)} \tag{10}$$

Where:
V is the vocabulary size (the number of distinct terms/words).
N is the total number of words in the text.

c) Maas:

$$a^2=\frac{\log(N)-\log(V)}{\log(N^2)} \tag{11}$$

$$\log V_0=\frac{\log V}{\sqrt{1-\frac{\log V^2}{\log N}}} \tag{12}$$

The term "logeV0" is equivalent to 'logV0", but it should be noted that the natural logarithm (with base e) is employed for the logarithmic computations. Furthermore, the computations incorporate the variables a, $\log(V_0)$ (which exhibit dissimilarity from their prior values), and V', which function as indicators of the proportional expansion of vocabulary across the text.

d) Mean segmental TTR(Type Token Ratio) (MSTTR):

$$\text{Mean Segmental TTR}=\frac{\text{Total TTR in all segments}}{\text{Number of segments}}$$

e) Moving Average TTR(Type Token Ratio) (MATTR): The formula remains the same as for MSTTR, however MATTR computes the average Type-Token Ratio (TTR) by considering a sliding window of words instead of mutually exclusive segments as in MSTTR.

f) Simpson's Index:

$$D=1-\sum_{i=1}^{V}(\frac{n_i}{N})^2 \tag{13}$$

Where:
V is the vocabulary size (the number of distinct terms/words).
n_i is the frequency of the ith word.
N is the total number of words in the text

B. Model Training and xAI Pipeline

Fig. 2 demonstrates the classification and xAI pipeline as a whole which includes training of the datasets on various machine learning models and then using xAI libraries like LIME and SHAP to get various insights regarding the data points.

1) Machine Learning and Classification: Both generated datasets are trained on classification models such as Logistic Regression, Decision Tree Classifier, Random Forest, Support Vector Machines, and Gradient Boosting. This is done to see which model will perform better on these data sets. The classification task is whether a given text is AI-generated or human-generated.
2) Top Model Selection: The best two of the five models trained before are chosen for xAI analysis using LIME and SHAP as these models will provide better insights than the others. The top two models are selected on the basis of classification metrics such as accuracy,f1-score, precision, and recall.
3) xAI Analysis: Arbitrary data points of AI-generated are chosen and LIME and SHAP analysis is implemented. Model weights of the two best machine learning algorithms are used. The same is done for human-generated text as well. How LIME and SHAP were used in this research is discussed later.
4) Identifying Common Patterns: Upon conducting these xAI techniques, the ensuing analysis reports offer a wealth of insights. These insights are subsequently juxtaposed, whereupon commonalities amongst the patterns identified by various models are isolated. This convergence of identified features in both AI and human-generated text serves as a critical juncture, underpinning the suggestion of a preferred technique for discerning between AI-generated and human-generated text.

C. xAI

1) LIME(Local Interpretable Model-agnostic Explanations): LIME is effectively implemented to interpret models used to classify between AI-generated and human-generated text in a dataset. In this scenario, the goal is to understand how the model distinguishes between texts created by artificial intelligence systems and those written by humans. LIME can provide insights into which features or patterns the model relies on for making such distinctions.

In this research, LIME is implemented on various test data points to check which features were chosen by a particular model for classification. To implement LIME, a subset of data points is chosen randomly, comprising both AI-generated and human-generated text samples. Then LIME is used to identify the prevailing scores and metrics, and subsequently patterns congruous to AI as well as human-generated texts are studied and determined. These patterns are discussed later in this research.

2) SHAP (SHapley Additive exPlanations): SHAP is a powerful method that, like LIME, is used to interpret classification models for human-generated and AI-generated text. It helps in figuring out which parts of the text or words are most important to model's prediction of whether a piece of text was written by a person or an AI system i.e. it can be used to reveal the important factors influencing the model's conclusions when classifying AI-generated and human-generated text.

TABLE III. HYPERPARAMETERS TUNED FOR ORCA AND GPT-J DATASET

Models	**Hyperparameters tuned and tuned values for Orca dataset**	**Hyperparameters tuned and tuned values for GPT-J dataset**
Logistic Regression	'C' (Regularisation Strength):90.68	'C' (Regularisation Strength): 33.27
Random Forest	n_estimators': 212, 'max_depth': 30,'min_samples_leaf': 2, 'min_samples_split': 4, 'bootstrap': False	n_estimators': 761, 'max_depth': 35, 'min_samples_leaf': 8, 'min_samples_split': 4, 'bootstrap': False
Gradient Boosting	'n_estimators': 351, 'learning_rate': 0.18896, 'max_depth': 5, 'min_samples_leaf':8, 'min_samples_split': 8	'n_estimators': 486, 'learning_rate': 0.20436, 'max_depth': 9, 'min_samples_leaf': 7, 'min_samples_split': 3
SVM	'C' (Regularization Parameter): 9.05764, 'kernel': 'linear', 'degree': 2, 'gamma': 0.01	'C'(Regularization Parameter): 5.185706911647028, 'kernel': 'rbf', 'degree': 3, 'gamma': 0.1

To use SHAP, a similar method of choosing a subset of data points is used that includes both types of text examples. SHAP then generates perturbed versions of these data points, similar to what LIME does. But instead of fitting a separate model that can be understood, SHAP uses an idea from cooperative game theory called Shapley values. These numbers tell how much each feature contributes to the prediction for a certain instance. The following are some SHAP visualization techniques that are used in this research to analyze the importance of individual features in the model's decision process:

1) Summary Plots: These graphs show how important each trait is across the whole dataset. They show the Shapley values for each feature, which show how they make the model's prediction move away from the average (base) estimate.
2) Waterfall Plots: Waterfall plots show how the Shapley value of each feature adds to the final prediction for a single instance. It makes it easier to see how the contributions add up.
3) Force Plots: Force plots are made to show how predictions can be made for specific cases. They show how the value of each attribute and its Shapley value interact to affect the final prediction.

V. EXPERIMENTATION

A. Parameter tuning

All the models were subjected to hyperparameter tuning to optimize the performance of various machine-learning models for AI-generated text detection. The process involved systematically searching and evaluating different combinations of hyperparameters to give the best set of hyperparameters that maximized the model's accuracy. For each model, a range of hyperparameter values was specified and RandomizedSearchCV was used which effectively sampled and cross-validated these values. This meticulous process enabled the models to better capture the patterns, resulting in improved accuracy and predictive capabilities. Table III shows the various hyperparameters used and their respective "best" values for model training to boost the accuracy of the models.

TABLE IV. CLASSIFICATION METRICS FOR ORCA GENERATED DATASET

Model Name	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	**0.92**	**0.92**	**0.93**	**0.93**
Decision Tree	0.80	0.79	0.80	0.77
Support Vector Classifier	**0.91**	**0.92**	**0.93**	**0.92**
Random Forest	0.86	0.86	0.84	0.89
Gradient Boosting	0.89	0.89	0.88	0.90

TABLE V. CLASSIFICATION METRICS FOR GPT-J GENERATED DATASET

Model Name	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	0.67	0.68	0.65	0.66
Decision Tree	**0.78**	**0.76**	**0.75**	**0.76**
Support Vector Classifier	**0.71**	**0.71**	**0.69**	**0.70**
Random Forest	0.70	0.69	0.70	0.71
Gradient Boosting	0.65	0.65	0.64	0.61

From the above models, an ensemble model was created combining the predictions of various models using a weighted average based on their accuracy scores. The weights were determined by normalizing the accuracy scores, ensuring their sum equals 1.

VI. RESULTS AND DISCUSSION

A. Classification Results

Both the datasets, the ones generated by GPT-J and Orca were trained on the various classification models mentioned above — Logistic Regression, Decision Tree, Random Forest, Support Vector Classifier, and Gradient Boosting. Both datasets had varied accuracies for the models trained. The various classification metrics such as accuracy, F1-Score, precision, and recall [40] for both datasets are illustrated in Table IV and Table V.

B. xAI Results and Inferences

xAI was then implemented to determine which features are dominating and have a higher impact in determining the class label of a particular data point.

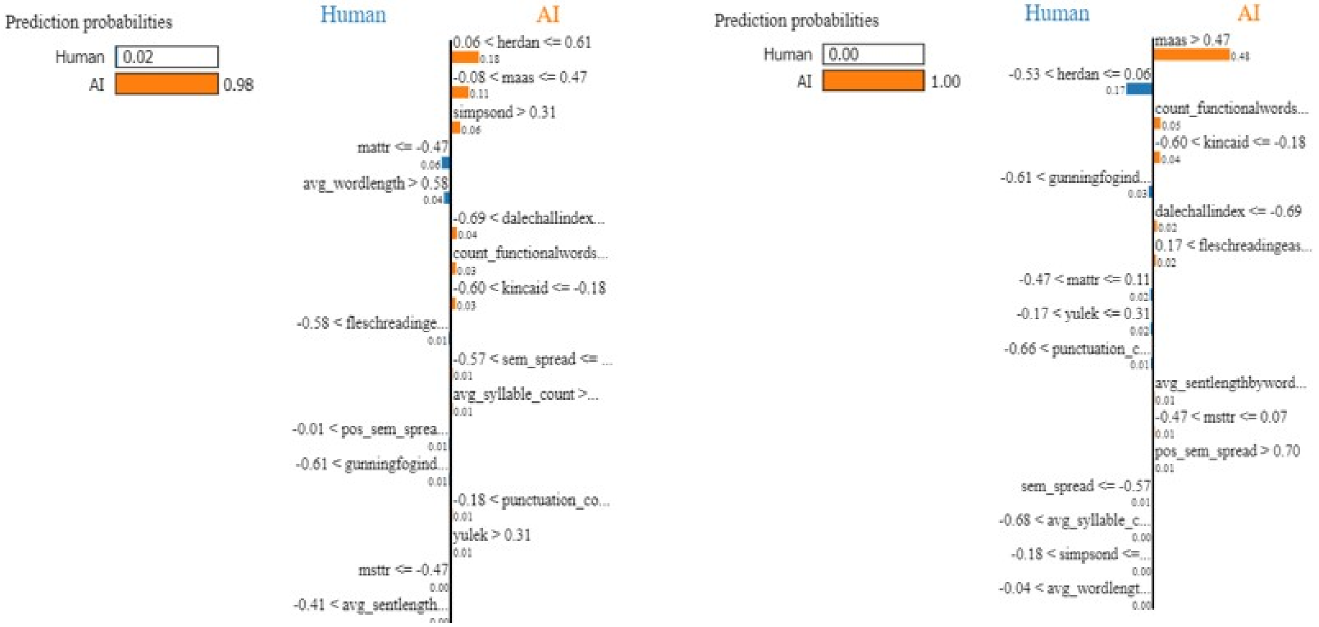

Fig. 3. LIME Interpretability Graphs for arbitrary AI-generated texts

LIME was then implemented on the Orca-generated dataset as it yielded better results to determine which features played an important role in how the model classified a data point as either human-generated or AI-generated text. This feature importance identification then helps in identifying what kinds of lexical and other features related to text are prominent in AI-generated and human-generated text.

The Fig. 3 has two LIME graphs for two separate AI-generated text data points. In the first image— on the left— LIME has Herdan's C [35], MaaS [36], and Simpson's index [38] have an abundant positive effect on the classification of this data point as an AI-generated text whereas MATTR and the average_word_length feature has a negative impact on the classification. From looking at various LIME graphs for data points being classified as AI generated the most dominant features were Herdans C, MaaS, and the Simpson's Index.

Herdan's C was one of the features which highly impacted the classification. Herdan's C metric is used to determine a text's vocabulary richness and diversity. It computes the proportion of unique words relative to the total number of words in the text. A higher Herdan's C value indicates a more diverse vocabulary, whereas a lower value indicates a repetitious or restricted vocabulary. It was observed from a sample of AI-generated data points that most of AI-generated text has a high value of this metric, meaning having a rich diversity of vocabulary was present. This is because the language model used to generate the data was pre-trained on massive datasets containing a wide range of text sources(3 billion parameters for Orca). But, there might be cases where the richness is abated because the dataset on which that language model was trained must not be up to standards.

The Simpson's Index can be used to measure the diversity of words in a given text within the context of text analysis. A higher Simpson's Index value would indicate less word diversity, indicating that a small number of words are repeated frequently. A lesser Simpson's Index value indicates greater word diversity or the use of a greater variety of words. Furthermore, for Simpson's Index, the values were high i.e. they were between 0.65 to almost reaching 1. This indicates that the phrases or words in the text are repeated often. Consequently, the diversity decreases. This is because AI models, particularly language models, can occasionally generate text that tends to reuse certain phrases or patterns, resulting in a relatively smaller vocabulary. These models may generate coherent text, but they may lack the inherent variability and creativity of human-generated text [41]. However, it's important to note that this can vary based on the specific AI model, the input data it was trained on, and the prompt given for text generation [42].

Fig. 4 illustrates a set of examples of LIME interpretability graphs for human-generated text. Many data points were interpreted and the results were mostly opposite to the ones inferred by the AI-generated ones. For instance, the Herdans C constant was relatively low for various human-generated text data points. It was either relatively low or it negatively influenced the classification.

Fig. 5, Fig. 6, and Fig. 7 illustrate the summary, waterfall, and force plots respectively generated by SHAP for the ORCA dataset. Table VI shows some feature values characteristic to AI and Human-generated texts.

TABLE VI. SOME FEATURE VALUES CHARACTERISTIC TO AI AND HUMAN-GENERATED TEXTS

Feature	AI-Generated	Human Generated
Herdan's C	0.9214	0.8901
Simpson's Index	0.016	0.013
MATTR	0.9548	0.9203
Maas	0.0180	0.0196
Flesch-Kincaid grade level	37.07	52.96
Gunning Fog Index	41.32	56.99

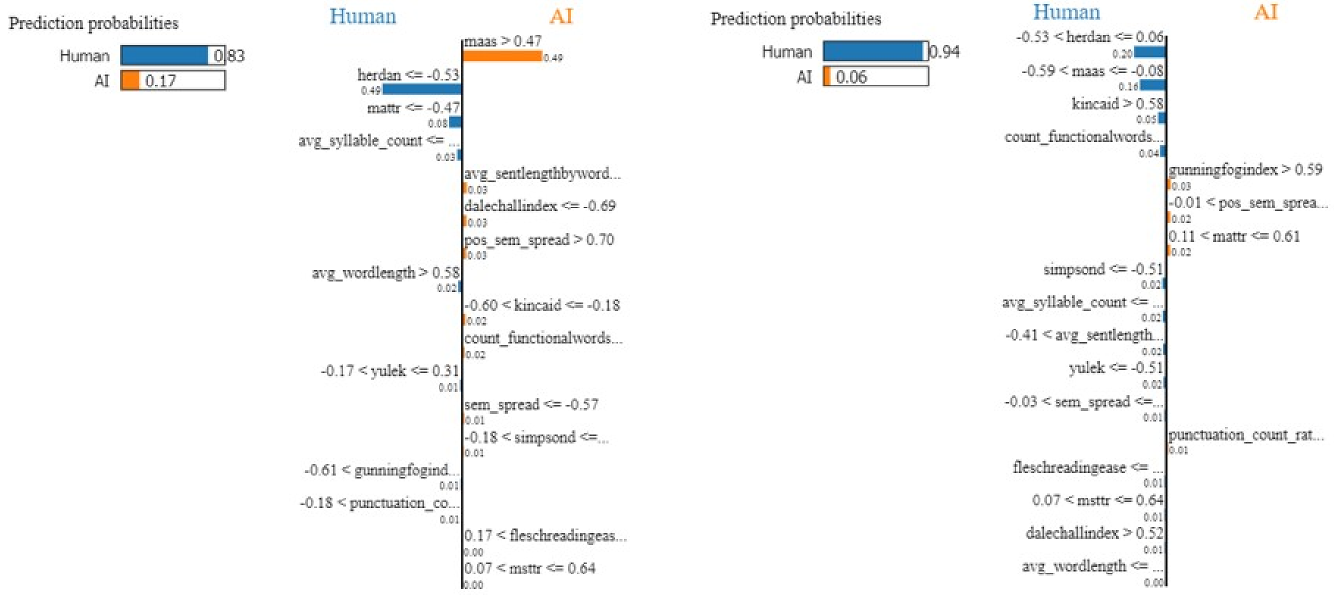

Fig. 4. LIME Interpretability Graphs for arbitrary human-generated texts

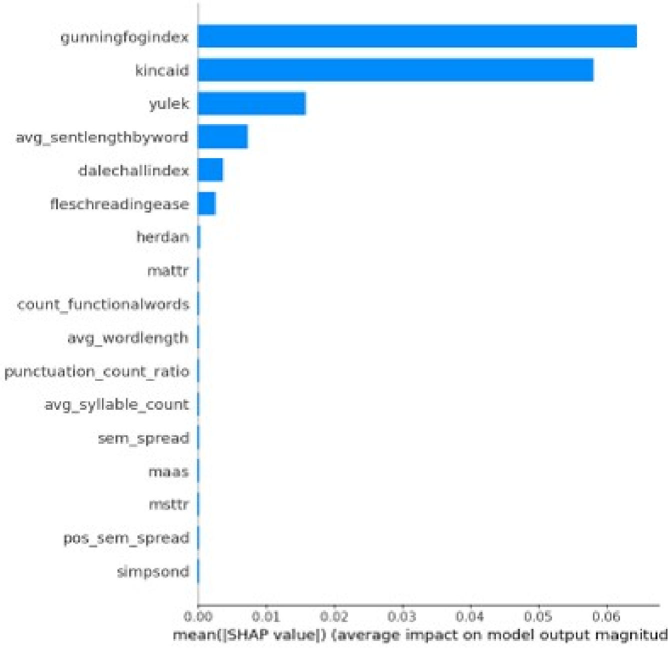

Fig. 5. Summary plot for SHAP values for Logistic Regression model trained on ORCA dataset

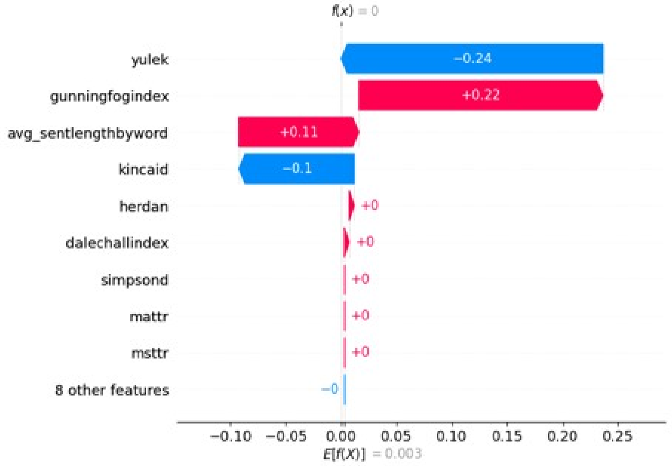

Fig. 6. Waterfall plot for SHAP values for Logistic Regression model trained on ORCA dataset

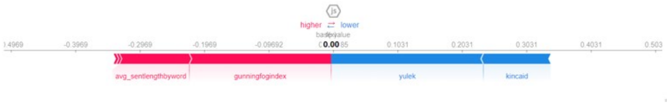

Fig. 7. Force plot for SHAP values for Logistic Regression model trained on ORCA dataset

VII. Conclusion

In conclusion, this research delves into the intricate realm of AI-generated text analysis and its differentiation from human-generated text. As Artificial Intelligence continues to revolutionize various facets of human activities, including text composition, the challenges of identifying AI-generated content have become increasingly pertinent due to concerns about misinformation, security vulnerabilities, and identity theft. The research methodology is multifaceted, combining linguistic analysis, readability metrics, semantic spread measurements, and vocabulary richness assessments to uncover essential textual attributes. By leveraging a composite ensemble of machine learning models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Classifier, and Gradient Boosting, the research demonstrates impressive efficacy with an accuracy of up to 93% in classifying AI-generated and human-generated text.

Moreover, the integration of xAI techniques, such as LIME and SHAP, provides invaluable insights into the features and patterns that influence the model's classification decisions. These insights reveal that certain attributes, such as Herdan's C, MaaS, and Simpson's Index, play pivotal roles in distinguishing AI-generated text from human-written content. These features highlight the richness of vocabulary, repetition of certain phrases, and syntactic patterns that are characteristic of AI-generated text.

The paper's limitation lies in its reliance on non-state-of-the-art models due to computational constraints, which may not fully represent the latest advancements in the field. These constraints, including limitations in computational resources and data availability, result in a performance gap compared to more advanced models. However, this limitation serves as a catalyst for future research that can harness the power of deep learning architectures and explainable AI (xAI) to delve into AI-generated text with greater sophistication. Additionally, it highlights the need to address ethical concerns and practical applicability as AI models evolve, making this paper a foundational stepping stone for deeper explorations in the future.

VIII. Future Scope

The research's future directions revolve around advancing AI-generated text analysis comprehensively. This entails harnessing larger and more diverse datasets spanning various domains, crucial for enhancing the detection system's real-world applicability. Alongside this, optimizing processing power to expedite analysis processes tied to xAI techniques like SHAP is essential. Incorporating additional style criteria such as linguistic tendencies and sentiment analysis aims to refine the methodology, deepening the grasp of distinguishing AI text styles from human language. This extends to evaluating a wider range of AI models beyond GPT-J and ORCA. Diversification of Machine Learning algorithms like K-Nearest Neighbors, Naive Bayes, and Neural Networks, as well as integration of Deep Learning algorithms like Transformer-based models, enhances AI text recognition. To fathom model decision-making, XAI methods like Grad-CAM and Integrated Gradients will be employed. Rigorous validation of authentic AI text data will test the proposed approach, collectively advancing differentiation between AI-generated and human-composed texts and propelling the field forward.

References

[1] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus *et al.*, "Artificial intelligence and life in 2030: the one hundred year study on artificial intelligence," *arXiv preprint arXiv:2211.06318*, 2022.

[2] P. S. Park, S. Goldstein, A. O'Gara, M. Chen, and D. Hendrycks, "Ai deception: A survey of examples, risks, and potential solutions," *arXiv preprint arXiv:2308.14752*, 2023.

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

[4] N. Maloyan, , B. Nutfullin, E. Ilyshin, and and, "DIALOG-22 RuATD generated text detection," in *Computational Linguistics and Intellectual Technologies*. RSUH, jun 2022. [Online]. Available: https://doi.org/10.28995%2F2075-7182-2022-21-394-401

[5] N. Fairclough, "Discourse and text: Linguistic and intertextual analysis within discourse analysis," *Discourse & society*, vol. 3, no. 2, pp. 193–217, 1992.

[6] G. Jawahar, M. Abdul-Mageed, and L. V. S. Lakshmanan, "Automatic detection of machine generated text: A critical survey," 2020.

[7] S. Gehrmann, H. Strobelt, and A. M. Rush, "Gltr: Statistical detection and visualization of generated text," 2019.

[8] R. Goebel, A. Chander, K. Holzinger, F. Lecue, Z. Akata, S. Stumpf, P. Kieseberg, and A. Holzinger, "Explainable ai: The new 42?" in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer International Publishing, 2018, pp. 295–303.

[9] H. Alamleh, A. A. S. AlQahtani, and A. ElSaid, "Distinguishing human-written and chatgpt-generated text using machine learning," in *2023 Systems and Information Engineering Design Symposium (SIEDS)*, 2023, pp. 154–158.

[10] G. Gritsay, A. Grabovoy, and Y. Chekhovich, "Automatic detection of machine generated texts: Need more tokens," in *2022 Ivannikov Memorial Workshop (IVMEM)*, 2022, pp. 20–26.

[11] C. H. Ng, H. S. Abuwala, and C. H. Lim, "Towards more stable lime for explainable ai," in *2022 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2022, pp. 1–4.

[12] C. Tantithamthavorn, J. Jiarpakdee, and J. Grundy, "Explainable ai for software engineering," 2020.

[13] S. Chakraborty, A. S. Bedi, S. Zhu, B. An, D. Manocha, and F. Huang, "On the possibilities of ai-generated text detection," 2023.

[14] D. Beresneva, "Computer-generated text detection using machine learning: A systematic review," vol. 9612, 06 2016, pp. 421–426.

[15] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, "Automatic detection of generated text is easiest when humans are fooled," 2020.

[16] R. Shijaku and E. Canhasi, "Chatgpt generated text detection," 01 2023.

[17] E. Vanmassenhove, D. Shterionov, and M. Gwilliam, "Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation," 2021.

[18] O. K. Kisselev and M. Aleksandr Kopotev, "A unified approach to interpreting model predictionsinvestigating lexical progression through lexical diversity metrics in a corpus of french l3," 2022.

[19] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 12 2017.

[20] S. Chowdhury and M. Schoen, "Research paper classification using supervised machine learning techniques," 10 2020, pp. 1–6.

[21] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, "Evaluating large language models trained on code," 2021.

[22] R. E. Wright, "Logistic regression," 1995.

[23] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[24] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[25] S. Suthaharan and S. Suthaharan, "Support vector machine," *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pp. 207–235, 2016.

[26] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[27] R. J. Larsen, K. A. Mercer, and D. A. Balota, "Lexical characteristics of words used in emotional stroop experiments." *Emotion*, vol. 6, no. 1, p. 62, 2006.

[28] C. van der Lee and A. van den Bosch, "Exploring lexical and syntactic features for language variety identification," in *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 2017, pp. 190–199.

[29] A. C. Bale and D. Barner, "The interpretation of functional heads: Using comparatives to explore the mass/count distinction," *Journal of Semantics*, vol. 26, no. 3, pp. 217–252, 2009.

[30] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," 1975.

[31] S. Zhou, H. Jeong, and P. A. Green, "How consistent are the best-known readability equations in estimating the readability of design standards?" *IEEE Transactions on Professional Communication*, vol. 60, no. 1, pp. 97–111, 2017.

[32] D. Świeczkowski and S. Kułacz, "The use of the gunning fog index to evaluate the readability of polish and english drug leaflets in the context of health literacy challenges in medical linguistics: An exploratory study," *Cardiology Journal*, vol. 28, no. 4, pp. 627–631, 2021.

[33] L. P. Stocker, "Increasing the precision of the dale-chall readability formula," *Reading Improvement*, vol. 8, no. 3, p. 87, 1971.

[34] J. A. Smith and C. Kelly, "Stylistic constancy and change across literary corpora: Using measures of lexical richness to date works," *Computers and the Humanities*, vol. 36, pp. 411–430, 2002.

[35] S. Herdan and Y. Sharvit, "Definite and nondefinite superlatives and npi licensing," *Syntax*, vol. 9, no. 1, pp. 1–31, 2006.

[36] J. Treffers-Daller, "Measuring lexical diversity among l2 learners of french," *Vocabulary knowledge: Human ratings and automated measures*, vol. 47, 2013.

[37] N. Chipere, D. Malvern, B. Richards, and P. Duran, "Using a corpus of school children's writing to investigate the development of vocabulary diversity," in *Technical Papers. Volume 13. Special Issue. Proceedings of the Corpus Linguistics 2001 Conference*. Citeseer, 2001, pp. 126–133.

[38] S. Jarvis, "Capturing the diversity in lexical diversity," *Language Learning*, vol. 63, pp. 87–106, 2013.

[39] Y. Anand, Z. Nussbaum, B. Duderstadt, B. Schmidt, and A. Mulyar, "Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo," https://github.com/nomic-ai/gpt4all, 2023.

[40] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

[41] A. Daniele, C. Di Bernardi Luft, and N. Bryan-Kinns, ""what is human?" a turing test for artistic creativity," in *Artificial Intelligence in Music, Sound, Art and Design: 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 10*. Springer, 2021, pp. 396–411.

[42] N. C. Chung, "Human in the loop for machine creativity," *arXiv preprint arXiv:2110.03569*, 2021.