STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, May 2025, pp 2186–2204.
Published online in International Academic Press (www.IAPress.org)

A Machine Learning Framework for Identifying Sources of AI-Generated Text

Md. Sadiq Iqbal, Mohammod Abul Kashem

Department of Computer Science and Engineering, Dhaka University of Engineering Technology, Gazipur, Bangladesh

Abstract The rise of AI-generated text requires efficient identification methods to ascertain its origin. This research presents a comprehensive dataset derived from responses to various questions posed to AI models including ChatGPT, Gemini, DeepAI, and Bing, alongside human respondents. We meticulously preprocessed the dataset and utilized both manual methods such as Count Vector (CV), Bag of Words (BoW), and Hashing Vectorization (HV), as well as automated Deep Learning (DL) models like Bidirectional Encoder Representations from Transformers (BERT), Extreme Language understanding Network (XLNet), Enhanced Representation through Knowledge Integration (ERNIE), and Generative Pre-Trained Transformers (GPT) to convert text into features. These features are then used to train multiple Machine Learning (ML) classifiers, including Support Vector Machines (SVM), Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), Naive Bayes (NB), and Extreme Gradient Boosting (XGB). This research also uses Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to maximize the classification accuracy of ML models. Remarkably, the combination of HV with LDA and XGB achieved the highest accuracy of 99.40%. Further evaluation using precision, recall, f1 score, specificity with Confusion Matrix (CM) and Receiver operating characteristic (ROC) Curve confirmed its superior performance, while Explainable Artificial Intelligence (XAI) tools such as Shapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) techniques are employed to explain the model's outputs, ensuring transparency and interpretability.

Keywords Text Classification, Manual Extraction, Deep Learning, Feature Optimization, Explainable Artificial Intelligence, Natural Language Processing

DOI: 10.19139/soic-2310-5070-2225

1. Introduction

Text identification is the process of categorizing textual data into predefined categories or labels based on its content. It is a fundamental task in natural language processing (NLP) and machine learning (ML) where the goal is to predict the class of a text document, such as an email, article, or review, based on its features [1]. In today's era of advanced artificial intelligence (AI), the proliferation of AI-generated content poses a significant challenge. Distinguishing between human-generated and AI-generated content has become increasingly difficult, impacting various sectors such as education, institutions, businesses, and content creation industries. Businesses are increasingly turning to AI-generated content to enhance operations, with 56% utilizing AI for operational improvements, 51% for cybersecurity and fraud management, and 47% employing digital personal assistants. Additionally, 46% utilize AI for customer relationship management, 40% for inventory management, and 35% for content production [2]. Other sectors leverage AI for various purposes such as product recommendations (33%), accounting assistance and supply chain operations (30%), recruitment and talent sourcing (26%), and audience segmentation (24%). The integration of AI text across these sectors underscores the need for robust methodologies to navigate the complexities of AI-generated content. Despite the significant advancements in AI technology and

*Correspondence to: Md. Sadiq Iqbal (Email: sadiq.iqbal@bu.edu.bd). Department of Computer Science and Engineering, Dhaka University of Engineering and Technology. Gazipur-1707, Bangladesh

ISSN 2310-5070 (online) ISSN 2311-004X (print)

the projected multi-billion-dollar market growth, the challenge of identifying the source of AI-generated content remains. However, amidst this advancement, it remains crucial to distinguish human-generated text, underscoring the importance of preserving authenticity and transparency in communication and content creation [3, 4].

In recent years, numerous studies have concentrated on ML and deep learning (DL) for identifying the sources of AI-generated text. These solutions are crucial for AI-based text identification as they provide methods to distinguish between human-generated and AI-generated text accurately [16]. Such differentiation is essential for various applications, including content moderation, plagiarism detection, and ensuring the authenticity of information in digital communication channels. By leveraging advanced ML and DL algorithms, these solutions offer robust frameworks to enhance the reliability and trustworthiness of AI-generated text analysis in diverse contexts. Thus, this research aims to develop a robust method to recognize the source of AI-generated text.

This research creates a dataset comprising text from diverse sources of AI-generated content, subjecting it to various preprocessing stages. We then employ both manual and DL-based feature extraction methods to process the dataset, preserving the integrity of the text. Next, we utilize a variety of ML models for text identification. Each model underwent evaluation using techniques such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA). Moreover, our exploration into explainable artificial intelligence (XAI) has played a pivotal role in interpreting the results of these experiments, shedding light on the efficacy of both ML and DL techniques. The proposed methodology entails the development of a cutting-edge system that incorporates the following elements:

- Investigating the challenge of AI-generated content in text identification.
- Employing a methodology that integrates manual feature extraction, DL techniques, and ML feature optimization methods.
- Prioritizing transparency and trustworthiness through the utilization of XAI techniques.

Our study is structured into distinct sections, each serving a specific purpose. Section 2 delves into a comprehensive review of pertinent literature in the field. Section 3 outlines the materials and methodology employed in our study. Subsequently, Section 4 provides a detailed depiction of the results and their analysis. Following that, Section 5 provides a discussion of the study. Finally, Section 6 concludes our research by summarizing significant findings and elucidating their implications.

2. Literature Review

Oghaz et al. [5] utilized a customized BERT model named RoBERTa for the classification of human and ChatGPT-generated content and achieved an impressive accuracy of 99.10%. However, the study acknowledged a notable drawback arising from RoBERTa's resource-intensive nature, potentially hampering its practical usage, especially in settings with restricted computational capabilities. Li et al. [6] employed RoBERTa for Russian artificial text detection dialogue and achieved a moderate accuracy of 64.73%. Their research shed light on the challenges faced by text detectors in keeping pace with the evolving landscape of AI, indicating the necessity for continuous adaptation and improvement in detection methodologies. Márquez et al. [7] utilized artificial neural networks (ANN) and Naive Bayes (NB) algorithms for English and Spanish text classification and attained accuracies of 67.00% and 64.41% respectively. Despite their contributions, they encountered limitations in effectively detecting computer-generated texts, suggesting the exploration of alternative AI models for improved performance.

Abburi et al. [8] employed an ensemble with a voting classifier for AI text classification and achieved an accuracy of 75.10%. Their study revealed a limitation in generalizing to unseen texts, emphasizing the importance of enhancing and fine-tuning models for the task of binary classification between human and generated texts. Son et al. [9] explored the combination of fluency and noise features for computer-generated text classification and achieved a notable accuracy of 80.35%. However, their research highlighted a limitation in adapting to diverse text inputs, suggesting further evaluation of frequency features and enhancement of noise features to address this challenge effectively. Son et al. [10] combined frequency, complex phrase, and consistency features for identifying computer-generated text, and achieved a high accuracy of 98.00%. Nonetheless, they faced difficulty in capturing the full

Table 1. Comparative Analysis of Text Classification Methods

Authors	Method	Identification	Accuracy	Limitation	Future Work
Oghaz et al.	RoBERTa	Classification of ChatGPT-Generated Contents	99.10%	Limited by resource-intensive RoBERTa	-
Li et al.	RoBERTa	Russian Artificial Text Detection in Dialogue	64.73%	Detectors struggle with evolving AI	-
Márquez et al.	ANN and NB	English and Spanish Text Classification	67.00% and 64.41%	Limited performance in detecting computer-generated texts	Implementing different AI models
Abburi et al.	Ensemble with Voting Classifier	AI Text Classification	75.10%	Limited generalization to unseen texts	Enhancing and tuning models for human- or generated binary classification task
Son et al.	Combination of Fluency and Noise Features	Computer Generated Text Classification	80.35%	Limited adaptation to diverse texts	Evaluating frequency features and enhancing noise feature
Son et al.	Combination of Frequency and Complex Phrase Features	Computer-Generated Text Using Fluency and Noise Features	98.00%	Difficulty capturing human language complexity	Improving their method to identify machine-generated text in novels, news reports

complexity of human language, indicating a need for improvement in their methodology to identify machine-generated text across various text genres and contexts such as novels, news reports, and human-written documents.

3. Materials and Methodology

This study aims to maximize text prediction accuracy using an approach involving manual techniques, deep learning (DL), and machine learning (ML). An overview of the research procedure is shown in Figure 1, with overall details provided in sections 3.1 to 3.10.

3.1. Dataset

This research generated a dataset by posing various questions to several AI platforms, including ChatGPT, Gemini, DeepAI, and Bing, as well as to human respondents. By compiling the answers from these sources, we assemble a dataset containing 5,019 samples. Each sample is labeled according to its source. The dataset is meticulously balanced across all sources to ensure representatives and reliability. The set of questions used for both AI and human respondents is detailed, with Table 2 providing a sample of our created dataset.

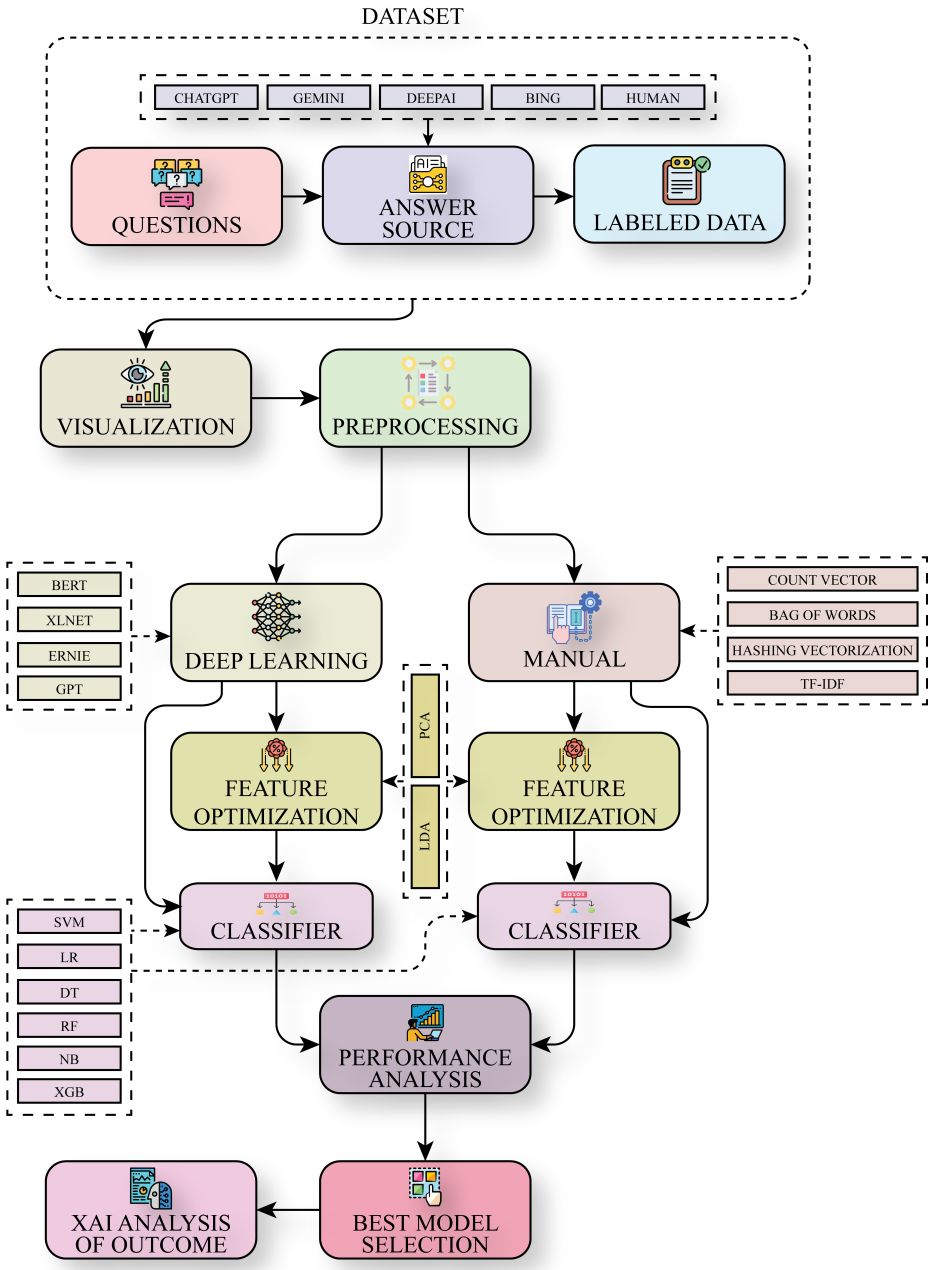

Figure 1. Text Identification Workflow of our Study.

3.2. Visualization

This section scrutinizes our created dataset using various visualization techniques. Employing visualization techniques such as Histograms and density plots, Box Plots, Violin Plots and Scatter Plots visualization specifically tailored for text data, we meticulously examine individual attributes. These graphical methods provide a holistic perspective of our dataset.

Table 2. Questions and Answers from Different Sources

Questions	Answer	From
What is temperature?	Temperature is a physical quantity that measures the degree of hotness or coldness of an object or substance relative to a reference point. It is usually expressed in units such as degrees Celsius (°C) or Fahrenheit (°F) and is used to describe the average thermal energy of the particles in a system. The temperature of an object can affect its properties, such as its state (solid, liquid, or gas), density, and thermal expansion. The temperature of a system can be increased or decreased by adding or removing heat, which causes the motion of the particles in the system to become more or less energetic.	ChatGPT
What is temperature?	Temperature is a physical quantity that expresses quantitatively the perceptions of hotness and coldness. Temperature is measured with a thermometer.	Human
What is photosynthesis?	Photosynthesis is a vital process for life on Earth. It is how plants produce the oxygen that we breathe and the food that we eat. Photosynthesis occurs in the leaves of plants, where sunlight is absorbed by a green pigment called chlorophyll. This energy is then used to convert water and carbon dioxide into oxygen and sugar.	Gemini
What is photosynthesis?	Photosynthesis is the process by which green plants and some other organisms use sunlight to synthesize nutrients from carbon dioxide and water.	Bing
What is computer vision?	Computer vision is a subfield of artificial intelligence that involves developing algorithms for processing visual information from images or videos to understand the world around us. Computer vision has many applications in fields such as robotics (navigation and manipulation), augmented reality (AR), virtual reality (VR), and autonomous driving (self-driving cars).	DeepAI

Figure 2 presents a comprehensive analysis of two dataset features using histogram and density plot visualizations. The histogram provides a clear depiction of the frequency distribution of values within discrete bins, while the density plot offers a smoothed representation, highlighting the density of occurrences across the data range [11, 12]. In the realm of NLP, these visualizations are instrumental in understanding textual datasets by revealing insights into linguistic features like word frequencies and text lengths. By leveraging these visualizations, we can discern prevalent trends and gain a nuanced understanding of the data characteristics, laying the groundwork for effective preprocessing and modeling strategies in NLP tasks.

Figure 3 showcases a detailed analysis of two key dataset features through box plot visualizations. A box plot offers a concise summary of the distribution of numerical data by displaying key statistical measures such as the median, quartiles, and outliers. The central box represents the interquartile range (IQR), with the median line

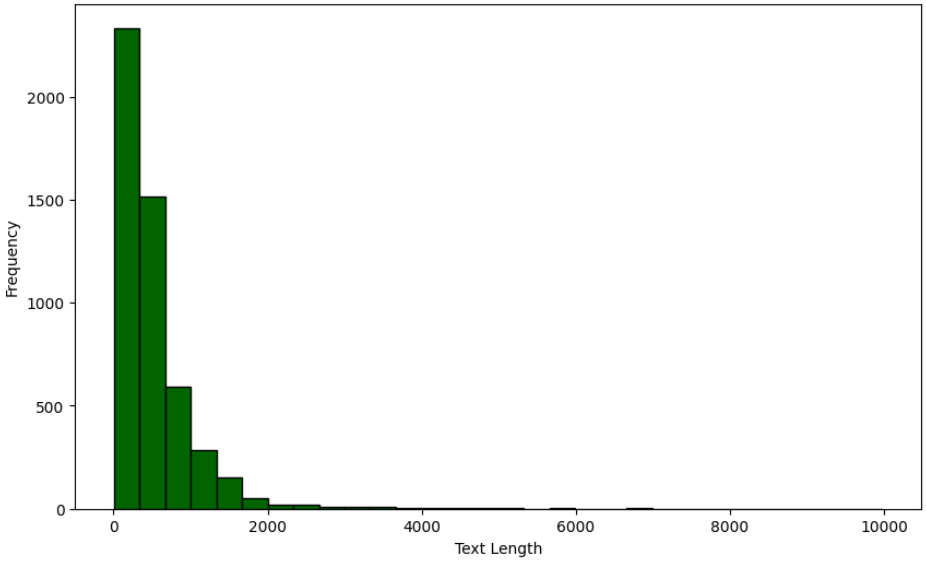

Figure 2. A comprehensive analysis of two dataset features using histogram and density plot visualizations.

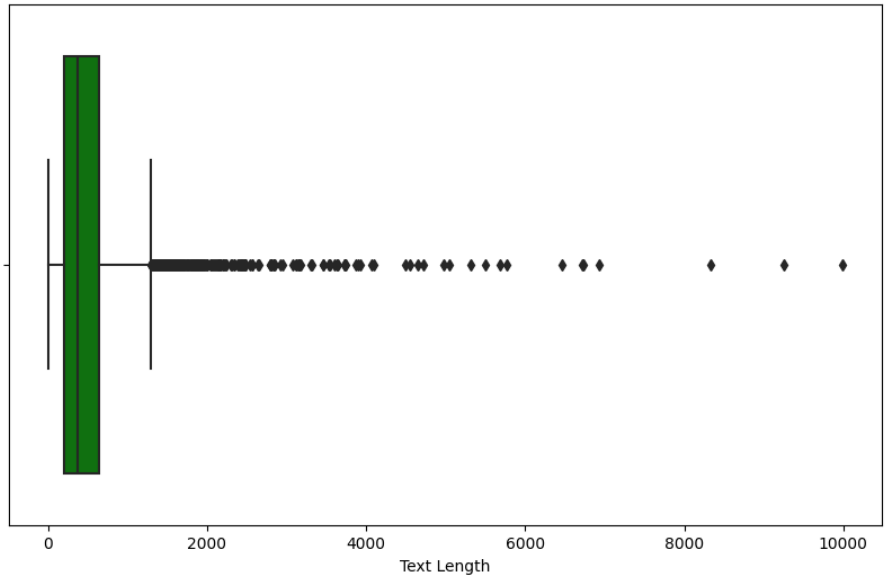

Figure 3. Box Plot of the text dataset

dividing it, while the whiskers extend to the minimum and maximum values within a specified range [13]. Box plots excel in identifying variations and outliers within the dataset, providing valuable insights into the data's spread and central tendency. In the context of NLP, applying box plot analysis to linguistic features like word frequencies or document lengths aids in understanding the variability and distribution patterns inherent in the text data. By leveraging box plot visualizations, we uncover significant trends, outliers, and variations, thereby informing preprocessing and modeling decisions crucial for effective text classification.

Figure 4 illustrates a violin plot that serves as a valuable visualization tool for analyzing text classification datasets. Combining aspects of a box plot and kernel density plot, it offers a detailed representation of the distribution of numerical data across different categories or classes [14]. Specifically, in the context of our study,

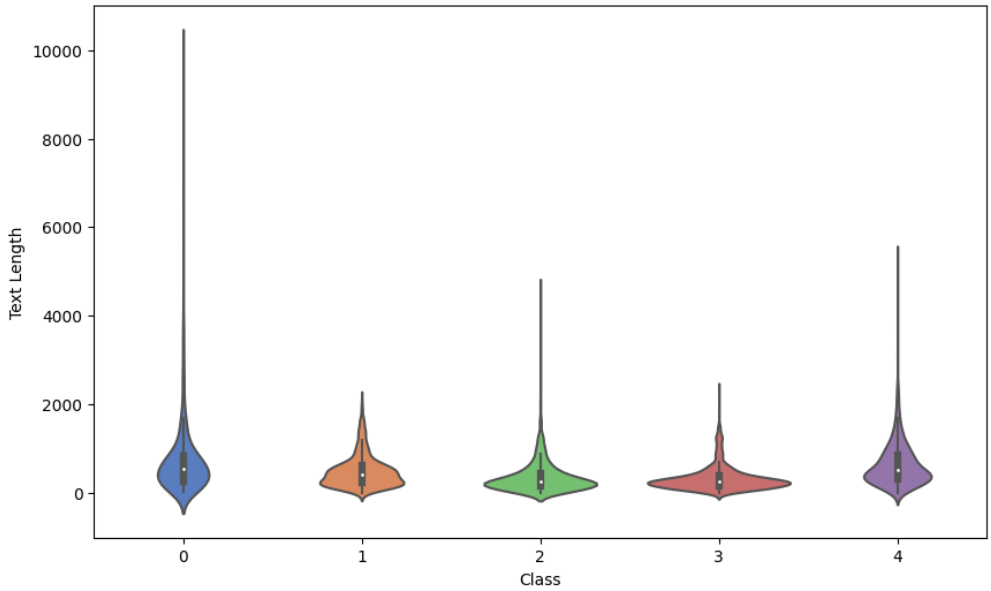

Figure 4. Violin Plot of the text dataset.

which focuses on text identification tasks, violin plots enable us to examine how various linguistic features, such as word frequencies or sentence lengths, are distributed across different classes or categories. By visually assessing the shape and spread of the violin plots for each class, we gain insights into the distribution patterns and identify any notable differences or similarities in text features among the classes. This understanding not only aids in feature selection and model training but also enhances the overall performance and effectiveness of our text classification algorithms.

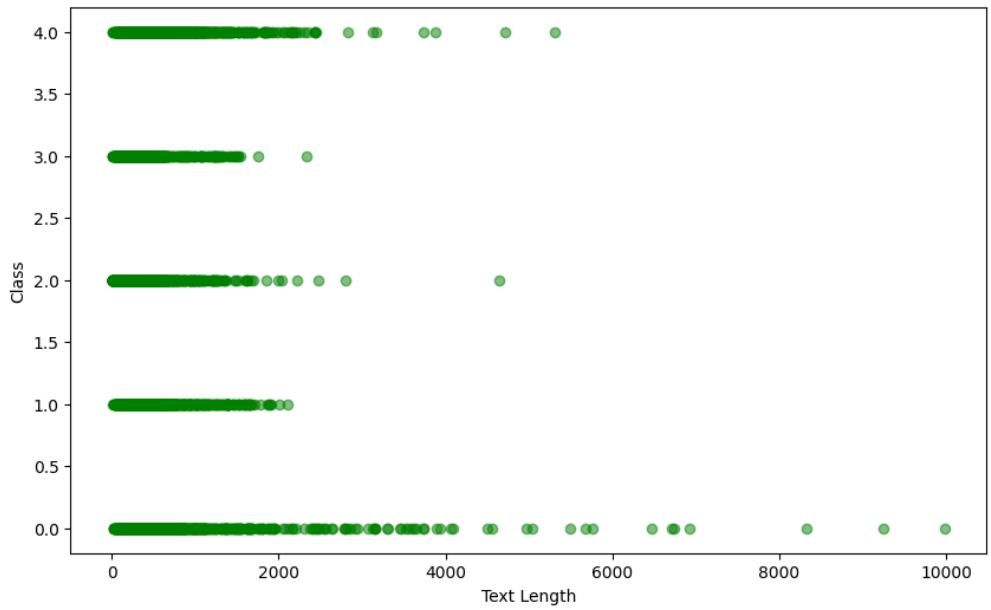

Figure 5. Scatter Plot of the text dataset.

Figure 5 shows a scatter plot, a visualization tool commonly used to explore relationships between pairs of numerical variables. Each data point on the plot represents a specific text feature or linguistic attribute, such as word frequency or document length, with one variable plotted along the horizontal axis and the other along the vertical axis [15]. In our text classification research, scatter plots are invaluable for understanding how different text features correlate with one another and influence classification outcomes. By examining the patterns and trends revealed in the scatter plot, we can gain insights into the relationships between text features and their impact on the classification process. This analysis guides us in selecting relevant features, optimizing model performance, and improving accuracy. Generating these visualizations helps us understand the dataset, including outliers, frequency density, text length, and data distribution. This understanding aids in preprocessing the data effectively and addressing outliers appropriately.

3.3. Preprocessing

In our dataset preprocessing phase, we encounter five missing values within a column. To ensure data integrity, we opt to address these missing values by dropping the associated samples, rather than imputing estimated values [17]. With this decision made, we transition to preparing our textual data for modeling. Given that our dataset contains various textual classes such as human, ChatGPT, Bing, Gemini, and Deep AI, we leverage word embedding techniques [18] to transform these textual classes into numerical representations. This transformation involves mapping each textual class to a numerical label such as 0 for human, 1 for ChatGPT, 2 for Bing, 3 for Gemini, and 4 for Deep AI. By employing word embedding, we enable our ML algorithms to comprehend and process these textual classes effectively throughout both the training and prediction phases, thereby enhancing the overall efficacy of our models. Additionally, to handle potential outliers, we applied LDA and PCA, which aid in reducing dimensionality while preserving the essential structure of the data, ensuring robustness and improving the performance of our models.

3.4. Deep Learning

In this stage, we perform feature extraction in our dataset, where we utilize DL techniques for extracting features from textual data. We apply several pre-trained models such as BERT [19], ERNIE [20], GPT-2 [21], and XLNet [22] to convert the text data into a numerical format understandable by ML algorithms. Subsequently, we apply ML to predict text classifications, leveraging the transformed numerical representations generated by the DL feature extraction process.

3.5. Manual

After executing the DL method, we applied manual feature extraction techniques including CV [23], BoW [24], TF-IDF [25], and HV [26] to convert textual data into numerical representations. These techniques facilitate the progression of our text classification task by transforming text data into a format compatible with further ML analysis.

3.6. Classifier

Utilizing various manual and DL feature extraction methods to convert our text data into numerical representations, we used a range of ML models including SVM [27], LR [28], DT [29], RF [30], KNN [31], NB [32], and XGB [33] to predict outcomes, employing various evaluation metrics. Then, we applied classification models to all the DL and manual models with each of the mentioned ML models to determine the optimal prediction model for text classification.

3.7. Feature Optimization

Feature optimization techniques refine and enhance the input features of a dataset to maximize the performance and efficiency of ML models. These techniques reduce redundancy, improve computational efficiency, and help models focus on the most relevant aspects of the data. Feature optimization is crucial in NLP tasks because text

data often contains high-dimensional and sparse features. In this study, we use two feature optimization techniques: LDA and PCA. Both methods aim to reduce the dimensionality of textual features while preserving the essential information required for accurate classification. LDA is a probabilistic topic modeling technique that identifies latent topics in a corpus of text. It assumes each document is a mixture of topics and each topic is a distribution of words. By analyzing these distributions, LDA reduces the dimensionality of textual data while representing the most important thematic structures [41]. PCA is a linear dimensionality reduction technique that identifies principal components to represent the variability in data. It transforms the data into a new set of uncorrelated variables (principal components), ranked by the amount of variance they explain [42].

3.8. Performance Analysis

This study employs common ML performance metrics, such as the CM, accuracy, precision, recall, F1 score, specificity, and ROC curve, to evaluate the model's performance. The CM is a table that represents the performance of a classification model, distinguishing between true and false outcomes for both actual and predicted results. Figure 6 shows the typical format of the CM used in this research. In Figure 6, this figure represents a confusion matrix for five classes: Human, ChatGPT, Bing, Gemini, and DeepAI, showing the predicted versus true class labels. Each cell indicates the count of True Positives (TP), False Positives (FP), and False Negatives (FN) for the respective class predictions. Human is correctly predicted as TP, with all other predictions as FP. Similarly, ChatGPT, Bing, Gemini, and DeepAI have their TP cells, while all other predictions for these classes are FP, with instances of other classes incorrectly predicted as these classes marked as FN. This matrix evaluates the classification performance, highlighting the accuracy and misclassifications for each class. We included a confusion matrix, which is detailed in the results section. Table 3 provides an overview of these performance metrics.

Table 3. Performance Evaluation Metrics for Text Classification Models

Metrics	Equation	Meaning
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN} \times 100$	Correctness percentage of classification model.
Precision	$\frac{TP}{TP+FP} \times 100$	Precision measures the accuracy of positive predictions.
Recall	$\frac{TP}{TP+FN} \times 100$	Captures the proportion of true positives correctly identified by a model.
Specificity	$\frac{TN}{FP+TN} \times 100$	Calculates the accuracy of negative predictions.
F1-Score	$2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100$	Balances precision and recall.

The ROC curve offers a clear graphical representation of a binary classification [44] model's performance by illustrating the trade-off between the True Positive Rate (TPR) on the Y-axis and the False Positive Rate (FPR) on the X-axis across various thresholds. This visual tool enables the assessment of a model's effectiveness, with the Area Under the Curve (AUC) serving as a key metric that quantifies the model's ability to differentiate between classes. The ROC curve facilitates a comprehensive evaluation by balancing recall and specificity, providing a thorough insight into the model's classification performance [34] [35]. By including the ROC curve in our results section, we accurately determined our model's reliability and precision.

3.9. Best Model Selection

This research applies both manual and automated feature extraction methods, which utilized various ML models, including SVM, LR, DT, RF, KNN, NB, and XGB, on all derived features. Additionally, several feature optimization techniques, such as PCA [36] and LDA [37], were implemented to further refine the predictions. Notably, the highest accuracy is achieved using the HV and XGB models in conjunction with LDA. This outcome highlights the superiority of the XGB model when paired with feature optimization through LDA.

True Class \ Predicted Class	Human	ChatGPT	Bing	Gemini	DeepAI
Human	True Positive (TP)	False Positive (FP)	False Positive (FP)	False Positive (FP)	False Positive (FP)
ChatGPT	False Negative (FN)	True Positive (TP)	False Positive (FP)	False Positive (FP)	False Positive (FP)
Bing	False Negative (FN)	False Positive (FP)	True Positive (TP)	False Positive (FP)	False Positive (FP)
Gemini	False Negative (FN)	False Positive (FP)	False Positive (FP)	True Positive (TP)	False Positive (FP)
DeepAI	False Negative (FN)	False Positive (FP)	False Positive (FP)	False Positive (FP)	True Positive (TP)

Figure 6. Confusion Matrix for the Classification Model Evaluating Five Classes (Human, ChatGPT, Bing, Gemini, and DeepAI).

3.10. XAI Analysis of Outcome

Following all the performance measurements, this study generated and visualized the final analysis using XAI with the best model. In the domain of enhancing the interpretability and transparency of AI systems, XAI emerges as a pivotal component in elucidating research outcomes. Employing a variety of strategies, XAI aims to provide accessible explanations of AI predictions to humans, while also offering insights into the internal mechanisms and potential biases inherent in AI models. By bridging the gap between sophisticated AI algorithms and human users, these XAI techniques empower individuals to make informed decisions based on AI outputs, thus fostering trust and enabling defensible decision-making processes [38]. XAI tools, including SHAP plots and LIME plots [43], have been employed in this research to improve our comprehension of AI decision-making processes and ensure that the outcomes are reliable, comprehensible, and accurate. SHAP plots are visualization techniques used to interpret the output of ML models, particularly in the field of NLP and text classification. In the context of text classification, SHAP plots help understand the importance of each feature, often corresponding to individual words or tokens in the text, in contributing to the model's prediction. By tokenizing and preprocessing the text data, training a text classification model, and generating SHAP values for each feature, a SHAP plot can be created to visually represent the impact of each feature on the classification decision. Analyzing the plot allows for the identification of influential words or tokens and the detection of biases or patterns in the data, providing interpretable insights into the model's decisions in text classification tasks in NLP [39]. LIME plots are used to explain the predictions made by ML models in the context of text classification and NLP. LIME plots provide insights into how a model makes predictions for individual texts by highlighting the most influential words or tokens. They work by analyzing the local neighborhood around a specific text and perturbing the text to observe the impact on the model's output. These perturbations generate explanations that can be visualized through a LIME plot, offering interpretable insights

into how the model arrived at its predictions for specific texts. This can help improve transparency and trust in identifying the source of AI content models by providing explanations for their decisions [40].

Table 4 shows the procedure outlining the steps involved in identifying AI-generated text using the combination of feature extraction methods, ML classifiers, and dimensionality reduction techniques. It also includes model evaluation and interpretability methods.

Table 4. Procedure for Identifying AI-Generated Text Using ML and DL Techniques

Steps	Description
Input	Dataset with responses from AI models (ChatGPT, Gemini, DeepAI, Bing) and human respondents.
Output	Best-performing model and explainability insights.
Step 1: Initialization	Define AI models: *ChatGPT, Gemini, DeepAI, Bing*. Define manual feature extraction methods: *Count Vector (CV), Bag of Words (BoW), Hashing Vectorization (HV)*. Define DL models: *BERT, XLNet, ERNIE, GPT*. Define ML classifiers: *SVM, LR, DT, RF, NB, XGB*. Define dimensionality reduction techniques: *Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA)*. Define evaluation metrics: *Accuracy, Precision, Recall, F1-Score, Specificity*.
Step 2: Data Preparation	Collect responses from AI models and human respondents. Preprocess the dataset (e.g., tokenization, stopword removal, lemmatization).
Step 3: Feature Extraction	For each feature extraction method (*CV, BoW, HV, BERT, XLNet, ERNIE, GPT*), transform text data into feature vectors.
Step 4: Train ML Models	For each feature extraction method, for each train-test split (*90:10, 80:20, 70:30, 60:40*), for each ML classifier (*SVM, LR, DT, RF, NB, XGB*), train classifier using extracted features and evaluate classifier using *Accuracy, Precision, Recall, F1-Score, Specificity*.
Step 5: Optimize Performance	Apply dimensionality reduction techniques (*LDA, PCA*), retrain classifiers with reduced features, and identify the best-performing combination (e.g., *HV + LDA + XGB*).
Step 6: Evaluation	Generate the Confusion Matrix (CM) for the best model. Plot Receiver Operating Characteristic (ROC) Curve.
Step 7: Explainability	Use SHAP to identify global feature importance. Use LIME for local interpretability of model predictions.
Step 8: Output Results	Report the best-performing model (*HV + LDA + XGB*) and summarize evaluation metrics and interpretability insights.

4. Results

Table 5 presents a comprehensive comparison of identifying the source of AI content using both manual and DL approaches, evaluated across various performance metrics including accuracy, precision, recall, F1-Score, and specificity with XGB classifier (XGB is chosen for its excellent performance in accurately identifying the source of AI content, as indicated by the information provided in Table 6). Among manual methods, the Bag of Words technique exhibits the highest performance, balancing an accuracy of 67.50% and the highest F1-Score of 66.74%, while TF-IDF and HV demonstrate lower performance. Within the realm of DL models, ERNIE stands out as the most effective, boasting the highest accuracy of 59.00% and strong precision, recall, and F1-Score metrics, while XLNET lags with the lowest performance metrics across the board. This analysis highlights the delicate performance differences among various identification of the source of AI content, emphasizing the significance of choosing an appropriate method tailored to the specific task demands and dataset attributes.

Table 6 illustrates the performance for identifying the source of AI content utilizing both manual and DL techniques, and evaluated across key metrics including accuracy, precision, recall, F1-Score, and specificity

Table 5. Performance of Different Manual and DL Feature Extraction Techniques with XGB classifier for identifying the source of AI content.

Method	Technique	Accuracy	Precision	Recall	F1-Score	Specificity
Manual	Count Vector	67.50%	71.13%	66.23%	66.37%	64.56%
	Bag of Words	67.50%	71.88%	67.87%	66.74%	69.70%
	TF-IDF	60.97%	61.45%	60.44%	58.59%	62.45%
	Hashing Vectorization	55.00%	63.25%	55.35%	55.87%	57.11%
DL	BERT	52.50%	53.88%	51.89%	51.06%	58.88%
	ERNIE	59.00%	59.09%	59.00%	58.81%	58.10%
	GPT	56.00%	57.81%	56.00%	56.22%	53.00%
	XLNET	46.00%	47.02%	46.00%	46.16%	57.80%

with XGB classifier. Among manual methods, Utilizing LDA HV demonstrates the highest overall performance, achieving remarkable accuracy of 99.40% and precision of 99.40%, with equally impressive recall and F1-Score metrics. Conversely, BoW exhibits notably lower performance, with an accuracy of 47.31% and the lowest F1-Score at 47.00%. Within the domain of DL models, GPT stands out with exceptional accuracy of 99.20% and precision of 99.19%, while ERNIE and XLNET showcase comparatively lower performance metrics. ERNIE and XLNET achieve accuracy scores of 43.40% and 41.50%, respectively, with F1-Scores hovering around 43-46%.

Table 6. Performance of Different Manual and DL Feature Extraction Techniques with feature optimization LDA and XGB classifier for identifying the source of AI content.

Method	Technique	Accuracy	Precision	Recall	F1-Score	Specificity
Manual	Count Vector	99.13%	99.14%	99.11%	99.12%	99.78%
	Bag of Words	47.31%	39.27%	37.53%	47.00%	81.25%
	TF-IDF	99.00%	99.00%	98.98%	98.98%	99.75%
	Hashing Vectorization	99.40%	99.40%	99.37%	99.38%	99.85%
DL	BERT	46.78%	47.09%	46.68%	45.44%	84.21%
	ERNIE	43.40%	43.70%	43.32%	43.81%	83.37%
	GPT	99.20%	99.19%	99.17%	99.18%	99.81%
	XLNET	41.50%	43.02%	42.00%	46.16%	82.80%

Table 7 presents the performance of various methods using the feature optimization technique PCA, evaluated across key metrics including accuracy, precision, recall, F1-Score, and specificity with XGB classifier. Among manual methods, the Bag of Words (BoW) technique demonstrates the highest overall performance, achieving an accuracy of 44.86%, a precision of 45.26%, and an F1-Score of 44.87%. Conversely, the Count Vector method exhibits lower performance metrics, with an accuracy of 37.42% and the lowest F1-Score at 30.89%. Within the domain of DL models, BERT stands out with the highest accuracy of 45.52% and precision of 45.86%, while GPT shows the lowest performance metrics, with an accuracy of 32.12% and an F1-Score of 36.23%. This comprehensive analysis highlights the variability in performance among different approaches to identifying the source of AI content, emphasizing the importance of selecting the appropriate method based on specific task requirements and dataset characteristics.

Based on the data presented in Tables 5 to 7, it's evident that HV with LDA emerges as the top performer. Table 8 provides a comprehensive comparison of different classifiers using HV and LDA. Notably, the XGB model showcases the highest overall performance, delivering exceptional results across all metrics. In contrast, the performance of other classifiers is notably lower. The RF classifier demonstrates the next best performance among traditional methods. SVM and LR yield moderate results, while KNN and NB exhibit the lowest performance. This comparison underscores the superior performance of the XGB model.

Figure 7 shows the LIME plot for our best model, which is XGB with HV and LDA. This plot indicates that in our model, the highest impacting class is 4, or DeepAI, which has the most significant influence in our identifying

Table 7. Performance of Different Manual and DL Feature Extraction Techniques with feature optimization PCA and XGB classifier for identifying the source of AI content.

Method	Technique	Accuracy	Precision	Recall	F1-Score	Specificity
Manual	Count Vector	37.42%	40.94%	38.34%	30.89%	82.00%
	Bag of Words	44.86%	45.26%	44.94%	44.87%	83.74%
	TF-IDF	36.22%	36.21%	36.35%	36.13%	81.57%
	Hashing Vectorization	43.26%	43.42%	43.24%	43.27%	83.31%
DL	BERT	45.52%	45.86%	45.60%	45.53%	83.90%
	ERNIE	37.40%	36.85%	37.16%	36.59%	81.85%
	GPT	32.12%	31.45%	33.22%	36.23%	80.32%
	XLNET	38.32%	41.84%	39.24%	35.79%	81.50%

Table 8. Performance Evaluation of Various Classifiers for Identifying the Source of AI Content

Classifier	Accuracy	Precision	Recall	F1-Score	Specificity
SVM	38.24%	38.50%	38.23%	37.91%	84.56%
LR	36.33%	36.39%	36.27%	37.31%	84.28%
DT	33.26%	33.39%	33.27%	33.19%	83.31%
RF	44.23%	45.53%	44.22%	43.98%	86.23%
KNN	25.81%	24.72%	25.67%	22.70%	81.18%
NB	21.31%	24.68%	21.14%	18.52%	80.31%
XGB	99.40%	99.40%	99.37%	99.38%	99.85%

the source of AI content study. Additionally, other classes like ChatGPT, Bing, and Gemini also visualize their impact using this SHAP plot.

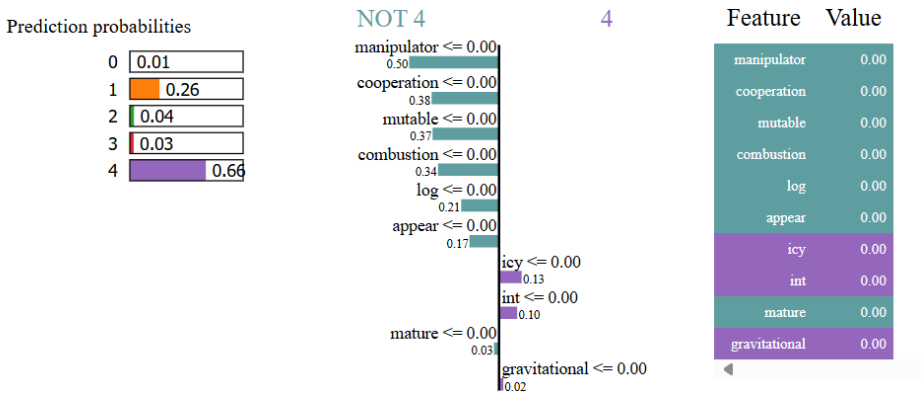

Figure 7. LIME Plot for the XGB Model with HV and LDA

Figure 8 depicts the SHAP plot for our optimal model, XGB with HV and LDA. This plot shows that the most influential class in our model is Human, which has the most impact on our identifying the source of AI content analysis. Furthermore, other models like ChatGPT, DeepAI, Bing, and Gemini also use SHAP plots to visualize their impact, providing a clear and comprehensive understanding of feature importance.

Figure 9 shows the ROC curve for our XGB model with Hashing Vectorization and LDA. The ROC curve plots the true positive rate against the false positive rate for each class, illustrating the model's performance in distinguishing between the different classes. The ROC curves for all classes (0, 1, 2, 3, 4) and the micro-average ROC curve all exhibit an area under the curve (AUC) of 1.00, indicating perfect classification performance. This

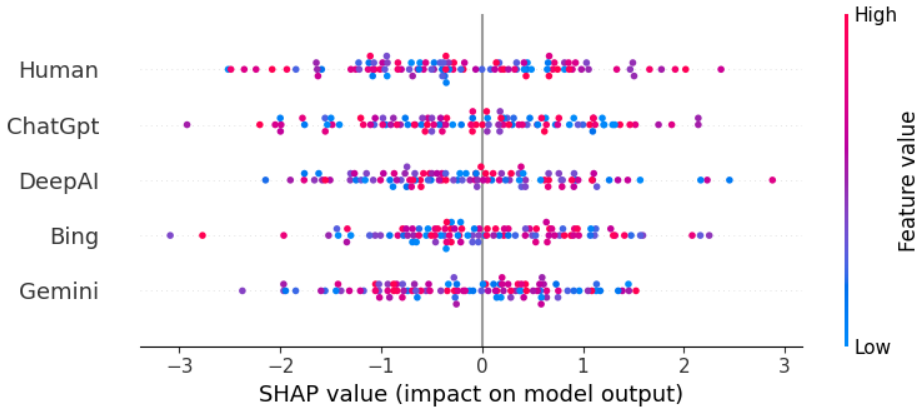

Figure 8. SHAP Plot for the XGB Model with HV and LDA

demonstrates that the model achieves an ideal balance between sensitivity and specificity, accurately distinguishing between classes with no false positives or false negatives.

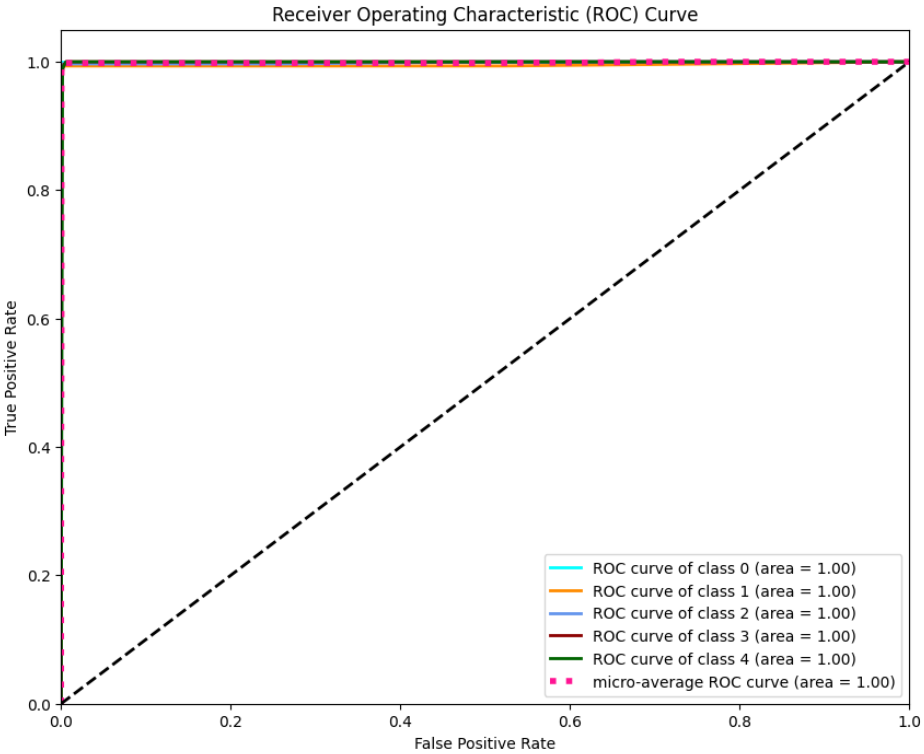

Figure 9. ROC Curve for the XGB Model with Hashing Vectorization and LDA

Figure 10 shows a confusion matrix (CM) illustrating the classification results for five classes: Human, ChatGPT, Bing, Gemini, and DeepAI. The matrix reveals high accuracy for each class, with very few misclassifications overall. Specifically, Human instances are perfectly classified with 304 correct identifications and no misclassifications. ChatGPT is correctly identified 307 times, with minor misclassifications as Gemini 1 and DeepAI 1. Bing shows 311 correct classifications with only 1 misclassified as DeepAI. Gemini has 273 correct

identifications but is misclassified once as ChatGPT and five times as DeepAI. DeepAI is correctly classified 302 times with no misclassifications into other classes. Overall, the matrix demonstrates excellent performance with only slight errors in distinguishing ChatGPT, Bing, and Gemini.

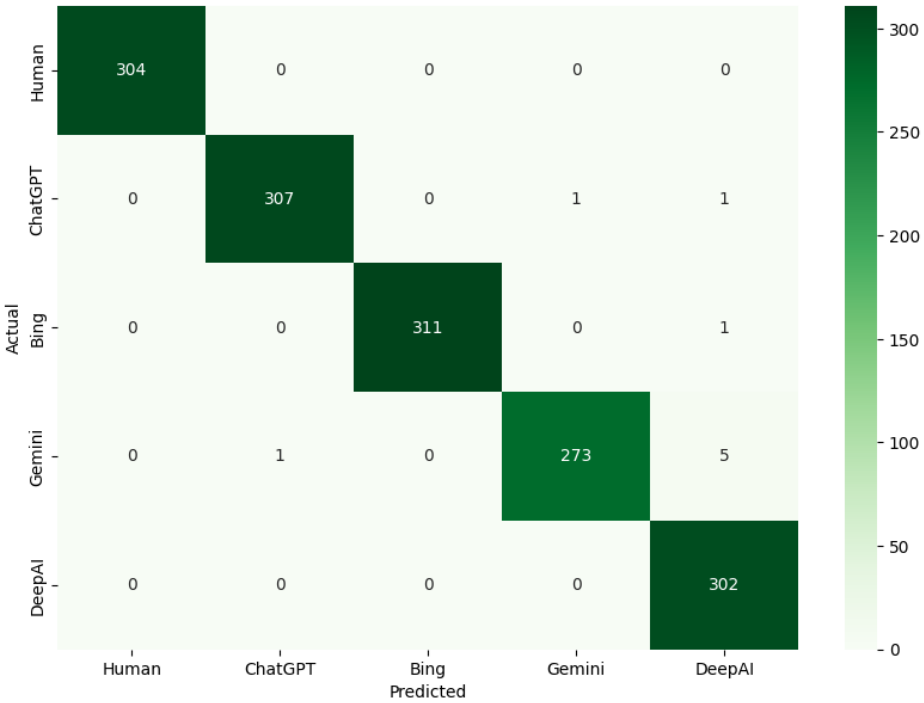

Figure 10. Confusion Matrix for Classification Results

4.1. External Evaluation

In this subsection, we test our best model, XGB, with auto and manual feature extraction on another dataset. We create a diverse dataset by selecting 50 samples for each class—DeepAI, Gemini, Bing, and ChatGPT—from the dataset used in this study. To expand this question set of 50 samples per class, we generate additional answers from various AI sources, including Claude, Google's PALM, and Copilot, as well as responses from various books [45, 46, 47, 48] for the class Human. This process results in a total of 400 samples across 8 classes. We evaluate this dataset using our models, and Table 9 presents the accuracy, precision, recall, F1-score, and specificity. We apply LDA, as it consistently outperforms traditional method and PCA in our previous analyses. The results indicate that XGB with XLNet achieves the best performance on this dataset, with an accuracy of 96.78%.

Table 10 presents the test results of both the original and extended datasets under various train-test distributions using the best model, XGB, with the best feature extraction method, LDA. The table shows that the 80:20 train-test distribution consistently delivers the best performance for this study, achieving the highest accuracy, precision, recall, F1-score, and specificity across both datasets.

5. Discussion

In this study, the disparities in accuracies across the provided tables can be attributed to various factors. High accuracies, such as those exceeding 99%—notably in Table 6 with 99.13% for CV, 99.00% for TF-IDF, 99.40% for HV, and 99.20% for GPT—are often achieved by employing techniques like XGB combined with the feature

Table 9. Performance of Different Manual and DL Feature Extraction Techniques with feature optimization PCA and XGB classifier for identifying the source of AI content for the extended dataset.

Method	Technique	Accuracy	Precision	Recall	F1-Score	Specificity
Manual	Count Vector	41.23%	38.72%	40.58%	39.84%	81.47%
	Bag of Words	43.68%	44.01%	42.95%	43.32%	82.63%
	TF-IDF	37.91%	36.47%	37.20%	36.88%	80.92%
	Hashing Vectorization	45.02%	43.89%	44.15%	43.75%	83.12%
DL	BERT	44.25%	43.68%	44.72%	44.10%	83.47%
	ERNIE	38.52%	39.41%	38.67%	39.02%	82.15%
	GPT	33.67%	34.29%	33.98%	34.15%	80.72%
	XLNET	96.78%	95.43%	96.12%	95.68%	97.03%

Table 10. Performance of Different Methods on Original and Extended Datasets with Various Train-Test Splits

Dataset Used	Method	Train : Test Distribution	Accuracy	Precision	Recall	F1-Score	Specificity
Original Dataset	HV	90:20	85.40%	85.30%	85.25%	84.90%	85.65%
		80:20	99.40%	99.40%	99.37%	99.38%	99.85%
		70:30	75.80%	75.65%	75.45%	75.78%	75.86%
		60:40	50.20%	50.10%	50.22%	50.18%	50.25%
Extended Dataset	XLNET	90:20	78.90%	78.80%	78.81%	78.78%	78.93%
		80:20	96.78%	95.43%	96.12%	95.68%	97.03%
		70:30	88.50%	88.35%	88.48%	88.56%	89.03%
		60:40	67.25%	67.18%	67.28%	67.05%	67.28%

optimization method LDA, which effectively capture complex relationships in the data. Additionally, sophisticated feature optimization methods like TF-IDF and HV, coupled with robust models, contribute to elevated accuracies by better capturing underlying data semantics. Conversely, low accuracies, such as those under 40%, are noticeable in Table 7, with feature optimization PCA achieving 37.42% for CV, 36.24% for TF-IDF, 32.12% for GPT, and 37.40% for ERNIE. These lower accuracies are often observed with simpler models like NB or KNN, which struggle to handle the complexity of our datasets effectively. Furthermore, inadequate hyperparameter tuning and suboptimal classifier choices can also contribute to lower accuracies. These extreme differences underscore the importance of meticulous model selection and feature optimization strategies in achieving high performance in ML tasks.

This method demonstrates practical relevance in several real-world applications. For instance, it is deployable in content moderation systems to identify inappropriate or harmful content across platforms. Similarly, it is applicable in plagiarism detection, enabling the identification of copied or AI-generated text in academic and professional documents. In cybersecurity, this method assists in detecting phishing content, malicious scripts, or AI-generated fraud attempts, enhancing system defenses against emerging threats. These case studies illustrate the method's impact and utility across diverse domains. Deploying the proposed method in real-world scenarios involves several challenges. Computational requirements pose a significant consideration, particularly for models with high resource demands. Real-time constraints necessitate optimizing the system for quick processing, ensuring it meets practical performance standards. Additionally, integrating the method into existing systems requires careful planning to ensure compatibility, scalability, and minimal disruption to current workflows. Addressing these challenges is crucial to unlocking the full potential of the proposed method in practical applications.

Hyperparameter tuning for XGB, RF, and SVM is conducted using Grid Search to systematically evaluate parameter combinations and optimize performance [49]. For XGB, we tune parameters like learning_rate, max_depth, and n_estimators to control learning pace and tree complexity [50], while regularization parameters (alpha, lambda) are adjusted to prevent over-fitting [51]. For RF, key parameters such as n_estimators (number of trees), max_depth (tree depth), and max_features (number of features considered for splitting) are fine-tuned to balance accuracy and generalization [52]. In SVM, we optimize C (regularization strength), gamma (kernel

coefficient), and kernel type to maximize the margin and minimize misclassification. Regularization is explicitly used to penalize complexity, ensuring robust and generalizable models [53].

We include a detailed comparative analysis with a wider range of state-of-the-art methods, DL models and ensemble methods (e.g., RF and XGB). While we implement and evaluate DL techniques like LSTM and CNN, their accuracies remain lower compared to other approaches, highlighting their limitations for this specific task. In contrast, ensemble methods like RF and XGB consistently deliver superior performance, with XGB achieving the highest accuracy, making it a strong advantage in the study. This comprehensive analysis underscores the robustness and effectiveness of the proposed framework.

Table 11 presents a comparison of our study with several existing studies. Our study demonstrates a significant advancement in AI source detection, achieving a remarkable accuracy of 99.40%, which surpasses the 99.10% accuracy by Oghaz et al. and 98.00% by Son et al. Unlike previous studies that did not identify multiple AI sources, our research introduces a 4-class AI source detection system, distinguishing between ChatGPT, DeepAI, Gemini, and Bing. Additionally, our study incorporates feature optimization techniques (LDA and PCA) and employs XAI models, further enhancing the interpretability and effectiveness of our approach. This comprehensive improvement underscores the methodological innovations and practical benefits of our work, marking a substantial contribution to the field.

Table 11. Comparison of Previous Works with Our Study

Related Studies	Accuracy	Multiple AI Source Detection?	Feature Optimization	XAI
Oghaz et al. [5]	99.10%	No	No	No
Li et al. [6]	64.73%	No	No	No
Márquez et al. [7]	67.00% and 64.41%	No	No	No
Abburi et al. [8]	75.10%	No	No	No
Son et al. [9]	80.35%	No	No	No
Son et al. [10]	98.00%	No	No	No
Our Study	99.40%	Yes	LDA and PCA	Yes

6. Conclusion

Overall, this research addressed the challenge of identifying the source of AI-generated content by utilizing a created dataset and employing a range of preprocessing and classification techniques. We employ both manual and DL feature extraction methods to obtain the actual outcomes of our study. The combination of HV with LDA and XGB yielded the highest accuracy. The evaluation using CM and the ROC Curve further verified the superior performance of this approach. Additionally, the incorporation of SHAP and LIME techniques ensured the transparency and interpretability of the model's outputs. However, the study is limited by the dataset size and diversity, which may affect generalizability, and the computational complexity of certain methods. In future work, we plan to implement a fusion method for NLP identifying the source of AI content, combining both automated and manual methods, to potentially improve our results. Furthermore, we aim to explore the integration of this framework into real-time systems and test its robustness under diverse real-world conditions, such as noisy or incomplete data.

REFERENCES

1. Sammons, M., Christodoulopoulos, C., Kordjamshidi, P., Khashabi, D., Srikumar, V., & Roth, D. (2016, May). Edison: Feature extraction for nlp, simplified. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 4085-4092).
2. Matzelle, E. (2024, February 29). Top artificial intelligence statistics and facts for 2024. CompTIA. https://connect.comptia.org/blog/artificial-intelligence-statistics-facts (Accessed June 5, 2024).

3. Forbes Advisor. (2023). Top AI statistics and trends for 2024. Forbes. https://www.forbes.com/advisor/in/business/ai-statistics/ (Accessed June 5, 2024).
4. Duarte, F. (2024, January 29). AI market size statistics (2024). Exploding Topics. https://explodingtopics.com/blog/ai-market-size-stats (Accessed June 5, 2024).
5. Oghaz, M. M. D., Dhame, K., Singaram, G., & Saheer, L. B. (2023). Detection and Classification of ChatGPT Generated Contents Using Deep Transformer Models. Authorea Preprints.
6. Li, B., Weng, Y., Song, Q., & Deng, H. (2022). Artificial text detection with multiple training strategies. arXiv preprint arXiv:2212.05194.
7. Morales-Márquez, L. E., Barrios-González, E., & Pinto-Avendaño, D. E. (2023). Artificial Intelligence-Based Text Classification: Separating Human Writing from Computer Generated Writing.
8. Abburi, H., Suesserman, M., Pudota, N., Veeramani, B., Bowen, E., & Bhattacharya, S. (2023). Generative ai text classification using ensemble llm approaches. arXiv preprint arXiv:2309.07755.
9. Nguyen-Son, H. Q., & Echizen, I. (2018). Detecting computer-generated text using fluency and noise features. In Computational Linguistics: 15th International Conference of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon, Myanmar, August 16–18, 2017, Revised Selected Papers 15 (pp. 288-300). Springer Singapore.
10. Nguyen-Son, H. Q., Tieu, N. D. T., Nguyen, H. H., Yamagishi, J., & Zen, I. E. (2017, December). Identifying computer-generated text using statistical analysis. In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1504-1511). IEEE.
11. Scott, D. W. (2010). Histogram. Wiley Interdisciplinary Reviews: Computational Statistics, 2(1), 44-48.
12. Chen, Y. C. (2017). A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology, 1(1), 161-187.
13. McGill, R., Tukey, J. W., & Larsen, W. A. (1978). Variations of box plots. The american statistician, 32(1), 12-16.
14. Kenny, M., & Schoen, I. (2021). Violin SuperPlots: visualizing replicate heterogeneity in large data sets. Molecular Biology of the Cell, 32(15), 1333-1334.
15. McCarthy, D. J., Campbell, K. R., Lun, A. T., & Wills, Q. F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics, 33(8), 1179-1186.
16. Reddy, T., Williams, R., & Breazeal, C. (2021, March). Text Classification for AI Education. In SIGCSE (p. 1381).
17. Chaudhary, M., Saad, M., Nassar, L., & Karray, F. (2021, October). Evaluation of Imputation Models Based on the Enhancement to Yield Forecasting. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3416-3422). IEEE.
18. Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. APSIPA transactions on signal and information processing, 8, e19.
19. Gregory, P. A., Bert, A. G., Paterson, E. L., Barry, S. C., Tsykin, A., Farshid, G., ... & Goodall, G. J. (2008). The miR-200 family and miR-205 regulate epithelial to mesenchymal transition by targeting ZEB1 and SIP1. Nature cell biology, 10(5), 593-601.
20. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129.
21. Budzianowski, P., & Vulić, I. (2019). Hello, it's GPT-2–how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. arXiv preprint arXiv:1907.05774.
22. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32.
23. Goyal, R. (2021, October). Evaluation of rule-based, CountVectorizer, and Word2Vec machine learning models for tweet analysis to improve disaster relief. In 2021 IEEE Global Humanitarian Technology Conference (GHTC) (pp. 16-19). IEEE.
24. Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. International journal of machine learning and cybernetics, 1, 43-52.
25. Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing & Management, 39(1), 45-65.
26. Kanada, Y. (1990, March). A Vectorization Technique of Hashing and Its Application to Several Sorting Algorithms. In PARBASE (pp. 147-151).
27. Ma, Y., & Guo, G. (Eds.). (2014). Support vector machines applications (Vol. 649). New York: Springer.
28. LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.
29. Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.
30. Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25, 197-227.
31. Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.
32. Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).
33. Pathy, A., Meher, S., & Balasubramanian, P. (2020). Predicting algal biochar yield using eXtreme Gradient Boosting (XGB) algorithm of machine learning methods. *Algal Research*, 50, 102006.
34. Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, 507, 772-794.
35. Metz, C. E. (1978, October). Basic principles of ROC analysis. In *Seminars in Nuclear Medicine* (Vol. 8, No. 4, pp. 283-298). WB Saunders.
36. Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), 433-459.
37. Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis—a brief tutorial. *Institute for Signal and Information Processing*, 18(1998), 1-8.
38. Došilović, F. K., Brčić, M., & Hlupić, N. (2018, May). Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 0210-0215). IEEE.

39. Zhao, W., Joshi, T., Nair, V. N., & Sudjianto, A. (2020). SHAP values for explaining CNN-based text classification models. *arXiv preprint arXiv:2008.11825*.
40. Mardaoui, D., & Garreau, D. (2021, March). An analysis of LIME for text data. In *International Conference on Artificial Intelligence and Statistics* (pp. 3493-3501). PMLR.
41. Hossain, E., Alshahrani, A., & Rahman, W. (2023). News Modeling and Retrieving Information: Data-Driven Approach. Intelligent Automation & Soft Computing, 38(2).
42. Shaha, P., Khan, M. S. I., Rahman, A., Hossain, M. M., Mammun, G. M., & Nasir, M. K. (2024). A Prevalent Model-based on Machine Learning for Identifying DRDoS Attacks through Features Optimization Technique. Statistics, Optimization & Information Computing.
43. Chowdhury, S. H., Mamun, M., Hossain, M. M., Hossain, M. I., Iqbal, M. S., & Kashem, M. A. (2024, April). Newborn Weight Prediction And Interpretation Utilizing Explainable Machine Learning. In 2024 3rd International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE) (pp. 1-6). IEEE.
44. Swarna, R. A., Hussain, M. I., Iqbal, M. S., Mamun, M., & Chowdhury, S. H. ILF: A Quantum Semi-Supervised Learning Approach for Binary Classification.
45. Brookshear, J. G. (1991). Computer science: An overview. Benjamin-Cummings Publishing Co., Inc..
46. Leicester, H. M., & Klickstein, H. S. (1952). A source book in chemistry, 1400-1900 (Vol. 1). Harvard University Press.
47. Bunge, M. (2013). Foundations of physics (Vol. 10). Springer Science & Business Media.
48. Johnson, D., & Johnson, D. (2002). How to do everything with your digital camera. McGraw-Hill/Osborne.
49. Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv preprint arXiv:1912.06059.
50. Dalal, S., Onyema, E. M., & Malik, A. (2022). Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy. World Journal of Gastroenterology, 28(46), 6551.
51. Luketina, J., Berglund, M., Greff, K., & Raiko, T. (2016, June). Scalable gradient-based tuning of continuous regularization hyperparameters. In International conference on machine learning (pp. 2952-2960). PMLR.
52. Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: data mining and knowledge discovery, 9(3), e1301.
53. Al-Mejibli, I. S., Alwan, J. K., & Abd, D. H. (2020). The effect of gamma value on support vector machine performance with different kernels. Int. J. Electr. Comput. Eng, 10(5), 5497-5506.