STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, August 2025, pp 756–769.
Published online in International Academic Press (www.IAPress.org)

A Machine Learning Framework for Discriminating between ChatGPT and Web Search Results

Md. Sadiq Iqbal*, Mohammod Abul Kashem, Mohammad Asaduzzaman Chowdhury

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

Abstract ChatGPT is a large language model built by OpeanAI. It is based on an architecture called the Generative pre-trained transformer (GPT). It can generate text that appears to be written by a human and understands natural language questions. We want to investigate whether we can distinguish between query results from web search and ChatGPT by utilizing ML. To accomplish the investigation this research trains five different Machine learning (ML) methods on a balanced dataset containing 2010 samples of query results from ChatGPT and web search. These ML models are Random forest (RF), Naive Bayes (NB), Decision tree (DT), Support vector machine (SVM), and Logistic regression (LR). Each of these methods is experimented with two feature optimization techniques namely LDA and PCA. After analyzing the results of all experiments, it is determined that the combination of NB with LDA yields the highest accuracy of 99.75%. Besides this technique also identifies ChatGPT-generated and human-written text with an accuracy of 98.67 from an existing dataset, and this outcome outperforms the state-of-the-art (SOTA) techniques. However, the proposed intelligent approach will help to identify any text of ChatGPT.

Keywords ChatGPT, Classification, GPT, NLP, Machine Learning

DOI: 10.19139/soic-2310-5070-2338

1. Introduction

ChatGPT is a large language model built on the GPT architecture[1]. The GPT architecture is a type of neural network that utilizes a transformer-based approach. This approach is a preferred choice for many Natural language processing (NLP) tasks due to its superior performance compared to traditional Convolutional neural networks (CNNs) and Recurrent neural networks (RNNs) as well as in handling long-term dependencies and variable-length input sequences [2]. ChatGPT can carry out several NLP activities such as question-answering, text summarization, language conversion, and text generation. This model trains on a big text dataset of over 570GB from various sources, such as websites, books, articles, etc. [3, 4]. The process of locating information on the world wide web utilizing search engines like Google, Yahoo, Bing, etc. is referred to as web search. Web search is an essential aspect of information retrieval and plays a crucial role in everyday life [5]. However, the question-answering feature of ChatGPT now getting more popular day by day and is a replacement for web searches. Especially for academic purposes, people are using ChatGPT for a wide. So, this research tries to develop an intelligent technique to differentiate the result of a specific query received from both ChatGPT and web searches by utilizing intelligent techniques. This research investigates several works before going through the proposed approach. To recognize AI-generated text in the paper [6] the authors showed in detail theoretical explanation with experimented results for several methods. In paper [7] the authors proposed two different approaches based on text consistency to recognize the human-written as well as the mix of machine and human-written text. For machine-generated text,

*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)

this research utilized GPT-2 architecture. In the paper [8] the authors presented a method named DetectGPT which can identify text generated from a machine. This method used no trained classifier on any dataset rather it performed execution based on log probabilities and random perturbations of the text from a specific generic pre-trained language model. The researchers of paper [9] developed a method namely GLTR that could automatically detect generated text. GLTR was developed based on several statistical methods. To recognize both human- and ChatGPT-generated text, the authors of the paper [20] suggested the TSA-LSTMRNN model by combining the Tunicate swarm algorithm (TSA) with a Long short-term memory recurrent neural network (LSTMRNN). The method examined decision-making processes by extracting features using TF-IDF, word embedding, and count vectorizers. LSTMRNN provided higher performance with maximum accuracies of 93.17% and 93.83% for human- and ChatGPT-generated texts, respectively. In paper [21] ML strategy for distinguishing ChatGPT-generated text from human-authored content. The researchers examined eleven different algorithms for text classification using a Kaggle dataset of 10,000 messages, including 5,204 human-written ones from news and social media. The program attained a 77% accuracy rate when evaluating GPT-3.5-generated text. In their investigation, the authors [22] used a classification algorithm to automatically identify essays created using ChatGPT. For training and model evaluation, they used a dataset containing writings from both human writers and ChatGPT. The model, which was built on the XGBoost algorithm, successfully detected ChatGPT-generated text with a 96% accuracy rate in their method. From the analysis of prior works, we have found no method to distinguish between web search and machine-generated text. We have also found most of the methods used traditional probability or consistency techniques to recognize machine-generated text. Hence, this research tries to find a solution by utilizing ML to distinguish the text of web search and ChatGPT. The purpose of this research is to utilize several ML models (SVM, NB, RF, DT, and LR) to differentiate between query results from ChatGPT and a normal web search. We developed our dataset for this purpose. The dataset consists of the answers to random questions by ChatGPT and websites. To get efficient outcomes this research uses two feature reduction techniques PCA and LDA with all ML models. All experiments are performed with 10-fold cross-validation to evaluate how the models perform with the variation of the data. The experiments include the performance of ML models with and without feature optimization techniques. The outcome of the experiments shows that LDA with the mentioned ML models can able to differentiate the text of a query from ChatGPT and web search. The major contributions of this research are:

- Forming a dataset of 2010 data samples containing label text of the answers to random questions from ChatGPT and different websites.
- Comparing the ability of different ML techniques to recognize whether a text is from any website or ChatGPT.
- Analysis of the strength of feature optimization techniques PCA and LDA for ML-based text recognition.
- Outperforming the SOTA techniques with the proposed method to classify the ChatGPT generated and human-written text utilizing an existing dataset.

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. Materials and Methodology

Figure 1 depicts the working method of this research, and next sections describe it in detail. Algorithm 1 provides an overview of the details research at a glance.

Figure 1. Text Identification Workflow of our Study.

2.1. Dataset

This research creates a balanced binary dataset of 2010 data samples. The dataset holds the answers to random questions collected from both ChatGPT and random websites. The classes are marked as 'ChatGPT' and 'WebResult' in the dataset. Figure 2 presents the overview to create the dataset of this research.

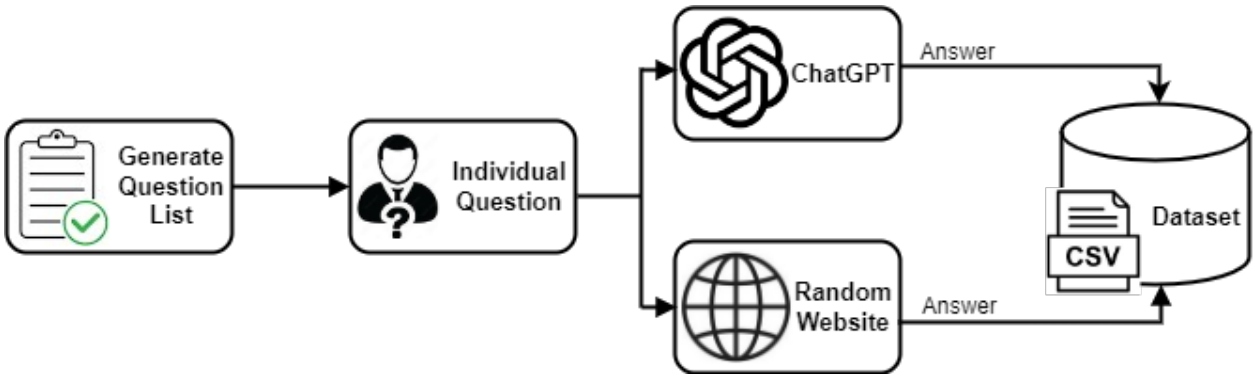

Figure 2. Text Identification Workflow of our Study.

The question list contains a total of 1005 questions on different subjects. Each of these questions is asked to both ChatGPT and a random website and the corresponding answers are marked as class 'ChatGPT' and 'WebResult' respectively. This research uses the web source selection criteria as:

- **Domains:** Sources were selected from reputable domains (e.g., `.edu`, `.gov`, `.org`, major news outlets).
- **Content Freshness:** Preference was given to content published within recent years to ensure relevance.
- **Query Diversity:** Questions spanned multiple categories—factual (e.g., "What is a VLAN?"), opinion-based (e.g., "Which accounting platforms have you worked on? Which one do you prefer the most?"), and technical (e.g., "Is coaxial cable still used in the computer network?").

Finally merging all the answers against the questions and corresponding classes in the CSV file the final dataset is formed. Table **??** presents the four samples of data from the dataset of this research. All of the SOTA method related to this research uses the ML technique to classify either a text is from ChatGPT or human writing. Thus, besides our dataset, we have also deployed the technique of the proposed method to an existing dataset labeled with ChatGPT generated or human writing text. The dataset is published by the Shijaku and Canhasi [22]. It is a balanced dataset that contains 126 texts for each category of ChatGPT and human written.

2.2. Preprocessing

During the execution of ML models, this research utilizes the 'Answer' and 'Class' attributes from the dataset, while the 'Question' attribute is dropped. For the 'Class' attribute, the values 'ChatGPT' and 'WebResult' are represented as 1 and 0, respectively. For the attribute 'Answer' we have used the Count vectorizer (CV) and TF-IDF [10] for tokenization and standard scaler (SS) [11] to standardize our dataset. However, CV with SS provides the best mechanism in our experiments.

Table 1. Examples of Questions, Answers, and Their Respective Classes

Question	Answer	Class
What is YouTube?	YouTube is a video-sharing platform that allows users to upload, share, and view videos. It was founded in 2005 and is now one of the largest and most popular websites in the world, with billions of users visiting the site every month to watch videos on a wide variety of topics, including music, entertainment, news, education, and more. YouTube is free to use, and anyone with a Google account can create a YouTube channel to upload and share their own videos. The site is also supported by ads, which allow creators to earn money from their content.	ChatGPT
What is YouTube?	YouTube is a free video-sharing website that makes it easy to watch online videos. You can even create and upload your own videos to share with others. Originally, created in 2005, YouTube is now one of the most popular sites on the Web, with visitors watching around 6 billion hours of video every month. If you've ever watched a video online, there's a good chance it was a YouTube video. For example, almost all of the video tutorials on our website are actually YouTube videos!	Other
What is NIC?	NIC (Network Interface Card) is a hardware component that connects a computer to a network. It provides the physical interface between the computer and the network and enables the computer to send and receive data over the network.	ChatGPT
What is NIC?	NIC is short for Network Interface Card. This is a peripheral card that is attached to a PC in order to connect to a network. Every NIC has its own MAC address that identifies the PC on the network.	Other

Count Vectorizer (CV) is a Natural Language Processing (NLP) tool for converting a collection of text documents into a token count matrix. Given a collection of N text documents $D = \{d_1, d_2, \ldots, d_N\}$, CV tokenizes each document and constructs a vocabulary $V = \{w_1, w_2, \ldots, w_M\}$ of all the unique terms w_i in D. In practice, CV applies various preprocessing steps, such as removing stopwords, stemming, and lemmatization, to minimize dimensionality and enhance the quality of the output matrix.

SS is a preprocessing technique that transforms each feature in such a way that it has a standard deviation of 1 and a mean of 0. It is based on the mathematical formula:

$$p = \frac{q - r}{\sigma} \tag{1}$$

where p is the standardized value, q is the feature's initial value, r is the average of the feature values, and σ is the feature values' standard deviation. SS is preferred because it standardizes data by removing the mean and scaling to unit variance, making it suitable for algorithms that assume normally distributed data. This technique ensures that all features contribute equally, preventing features with larger scales from dominating the model.

2.3. Feature Optimization

This research analyzes two feature optimization approaches namely PCA [12] and LDA [13] to exclude data redundancy, improve time efficiency, and minimize the number of input columns. Figure 3 shows the cumulative

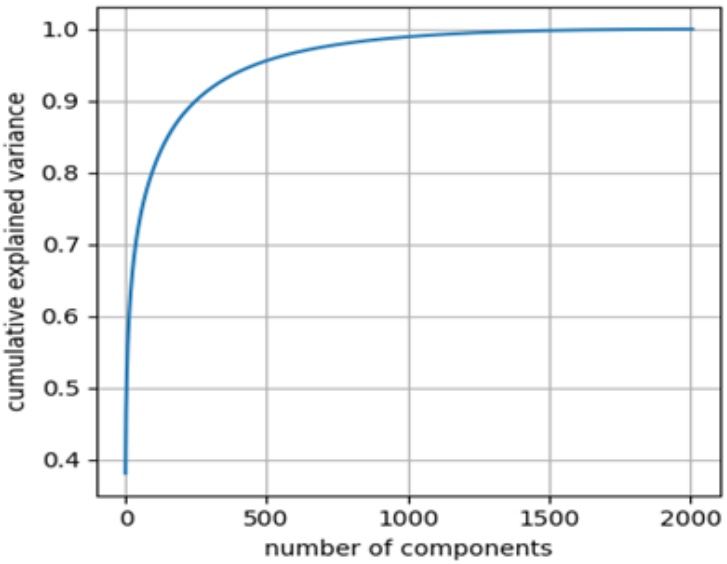

Figure 3. cumulative explained variance by PCA components

explained variance by PCA components, where initially, variance rises sharply with the number of components but then flattens after a few hundred, indicating that most information is captured early. This helps in selecting fewer components while retaining most data variability, making models faster and simpler. We did not create a similar graph for LDA because LDA components are limited by the number of classes minus one, meaning one component is available. Therefore, cumulative variance analysis is meaningful for PCA but not useful for LDA. Section 2.3.1 and 2.3.2 narrates the mechanism of PCA and LDA in detail.

2.3.1. PCA identifies the directions in the data with the highest variance, known as the principal components. These components are orthogonal to each other, meaning they are uncorrelated. The first principal component captures the highest variance, the second captures the second highest, and so on. The mathematical description of PCA is as follows:

- Determine the mean of each feature:
$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{2}$$
where μ_j represents the mean of the jth feature, n is the total number of data points, and x_{ij} refers to the value of the jth feature for the ith data point.
- Center the data by subtracting the mean from each feature:
$$\bar{x}_{ij} = x_{ij} - \mu_j \tag{3}$$
- Calculate the covariance matrix (Co) of the centered data:
$$Co = \frac{1}{n-1}\sum_{i=1}^{n}(\bar{x}_j \bar{x}_j^T) \tag{4}$$
where n is the number of data points, and T denotes the transpose operator.
- Compute the eigenvectors and eigenvalues of the covariance matrix:
$$Co\, v_i = \lambda_i v_i \tag{5}$$
where v_i is the ith eigenvector, and λ_i is the ith eigenvalue.

- Sort the eigenvectors in descending order of their eigenvalues and select the top k eigenvectors to form the projection matrix P:
$$P = [v_1, v_2, \ldots, v_k] \tag{6}$$
- Project the data onto the new k-dimensional space by multiplying the centered data matrix with the projection matrix:
$$Y = XP \tag{7}$$
where Y is the new k-dimensional dataset, X is the original dataset, and P is the projection matrix.

2.3.2. LDA is a popular ML approach for feature extraction and dimensionality reduction. LDA identifies the optimal linear feature combinations for discriminating between two or more classes. The mathematical description of LDA is as follows:

- Calculate the mean of each feature in the dataset for each class:
$$m_i = \frac{1}{n_i} \sum_{k=1}^{n} l_k \tag{8}$$
where m_i is the mean of the features for class i, n_i is the total number of data points in class i, and l_k is the kth data point in class i.
- Calculate the within-class scatter matrix S_w:
$$S_w = \sum_{i=1}^{c} \sum_{x_j \in C_i} (x_j - m_i)(x_j - m_i)^T \tag{9}$$
where c is the number of classes, C_i represents the ith class, and T denotes the transposition operator.
- Calculate the between-class scatter matrix S_b:
$$S_b = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T \tag{10}$$
where m is the mean of the overall dataset, m_i is the mean of class i, and n_i is the number of data points in class i.
- Find the eigenvalues and eigenvectors for the matrix $S_w^{-1} S_b$:
$$S_w^{-1} S_b v_i = \lambda_i v_i \tag{11}$$
where v_i is the ith eigenvector and λ_i is the ith eigenvalue.
- Sort the eigenvectors in decreasing order by eigenvalue and select the top r eigenvectors to construct the projection matrix P:
$$P = [v_1, v_2, \ldots, v_r] \tag{12}$$
- Obtain the new k-dimensional space of projected data by multiplying the centered data matrix by the projection matrix:
$$Y = XP \tag{13}$$
where Y is the new k-dimensional dataset, X is the original dataset, and P is the projection matrix.

2.4. Model Execution

All ML models of this research are trained and tested utilizing the mentioned dataset. This section describes the working mechanism of these models.

2.4.1. SVM in classification problems, finds the best hyperplane that separates the data into two groups [14]. Given a training set with inputs $x_1, x_2, \ldots, x_n$ and corresponding outputs $y_1, y_2, \ldots, y_n$, where $y_i \in$ {ChatGPT, WebResult}, SVM seeks a hyperplane that divides the data points into two groups. The hyperplane equation is given by:

$$w \cdot x + b = 0 \tag{14}$$

where w and b denote the weight vector and bias term, respectively.

The goal of SVM is to find the optimal values of w and b that best separate the data points. To achieve this, a mapping function is applied to transform the data into a higher-dimensional space. The transformed data is then used to find the hyperplane that separates the data points in this new space.

2.4.2. The working mechanism of RF can be broken down into the following steps [15]:

- Data Preparation: The first step is to separate the data into testing and training sets. The model is built using the training set, and its performance is evaluated using the test set.
- Tree Construction: The RF approach generates a large number of Decision Trees (DTs) from various subsets of the training data.
- Feature Selection: To identify the optimal split, a subset of features is randomly selected at each node of the DT. This helps to minimize tree correlation and increase tree variety.
- Split Selection: The optimal split is determined based on a specific criterion. This criterion measures the degree of homogeneity of the target variable within each branch of the tree.
- Tree Aggregation: Once all the trees are constructed, their predictions are combined by taking the majority vote (for classification) or the average (for regression) of their outputs. This produces a final prediction for each instance in the testing set.

2.4.3. DT is a ML technique that creates a decision-making model by generating a tree structure [16]. The DT algorithm is defined as follows:

Let X be a set of input features in a dataset D, and Y be a set of output labels. Let D be a dataset of n training examples, each consisting of a feature vector x_i and a corresponding label y_i:

$$x_i = [x_1, x_2, \ldots, x_m] \tag{15}$$

The goal of DT is to learn the function $f : X \to Y$, which maps input feature vectors (X) to output labels (Y). A decision tree represents the function f, with each internal node representing a decision based on a specific feature and each leaf node representing a label.

The decision tree is built iteratively by partitioning the training data into subsets based on the values of a selected feature. The feature that results in the optimal split is chosen based on a certain criterion (e.g., information gain). This process is repeated until a stopping condition is met, such as:

- A maximum tree depth is reached.
- All instances in a node belong to the same class.

2.4.4. NB Naive Bayes (NB) is a probabilistic ML approach that predicts using Bayes' theorem [17]. The general form of this theorem, as utilized in the NB model, is:

$$P(y \mid x_1, x_2, \ldots, x_n) = \frac{P(y) \times P(x_1 \mid y) \times P(x_2 \mid y) \times \cdots \times P(x_n \mid y)}{P(x_1, x_2, \ldots, x_n)} \tag{16}$$

where:

- $P(y \mid x_1, x_2, \ldots, x_n)$ is the probability of y given the values of features $x_1, x_2, \ldots, x_n$.
- $P(y)$ is the probability of y occurring in the dataset.
- $P(x_i \mid y)$ is the probability of x_i occurring given that y has occurred.
- $P(x_1, x_2, \ldots, x_n)$ is the marginal probability of features $x_1, x_2, \ldots, x_n$.

To predict the label (y) for a new feature, this probability equation is solved for each possible label, and the label with the highest likelihood is chosen.

2.4.5. LR Logistic Regression (LR) predicts the probability of an outcome based on one or more input features [18]. The sigmoid function is used by LR to convert the linear equation into a probability score ranging from 0 to 1. The sigmoid function is defined as:

$$f(x) = \frac{1}{1 + e^{-z}} \tag{17}$$

where x is the input feature, the linear combination of the input features and their weights is denoted by z, and e is Euler's number. The equation for the linear combination z is:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n \tag{18}$$

where β is the coefficient for each input feature, and n is the number of input features. LR estimates the coefficients (β) that reduce the discrepancy between probabilities and the actual binary outcomes. This is achieved by maximizing the log-likelihood function:

$$L(\beta) = \sum_{i=1}^{n} \left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right) \tag{19}$$

where n is the number of assessments, y_i is the binary outcome (e.g., ChatGPT, WebResult), p_i is the anticipated probability of the binary event, and $\log$ is the natural logarithm.

To estimate the coefficients (β), iterative numerical optimization techniques, such as gradient descent, are used by LR. These techniques seek coefficient values that minimize the log-likelihood function.

2.5. Performance Analysis

The dataset has 2010 samples after preprocessing of which 80% data is used for training and 20% for testing the ML models. To improve transparency, we have used 10-fold cross-validation. Accuracy, precision, recall, and F1 score are employed as the performance assessment metrics for the ML models [19]. Table **??** presents these metrics in detail. In table **??** TN, FN, FP, and TP present the number of True negative, False negative, False positive, and True positive predicted values by any ML model.

Table 2. Performance Evaluation Metrics for Text Classification Models

Metrics	**Equation**	**Meaning**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN} \times 100$	Percentage of correct predictions by the classification model.
Precision	$\frac{TP}{TP+FP} \times 100$	Accuracy of positive predictions.
Recall	$\frac{TP}{TP+FN} \times 100$	Proportion of actual positives correctly identified.
Specificity	$\frac{TN}{FP+TN} \times 100$	Accuracy of negative predictions.
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100$	Harmonic mean of precision and recall.

Algorithm 1 Working Procedure of Proposed Method

Input: Any text(T)
Output: Categories (ChatGPT/WebSearch)

Initialization:
ML models $(M_j) : \{RF, NB, DT, SVM, LR\}$
Feature optimization techniques $(F_k) : \{PCA, LDA\}$
Preprocessing techniques $(P_t) : \{CV, SS\}$
Training data $(L) : \{T_x, T_y\}$
Testing data $(I) : \{X_t, X_y\}$
Dataset (D)
Result comparer (P)
End Initialization

Start
Apply P_t to D
Split D into 10-fold cross-validation with ratio $L : R = 8 : 2$
For each fold i
  For each M_j, where $j = 1, 2, 3, 4, 5$
    Train M_j using $\{T_x, T_y\}$
    Evaluate M_j using $\{X_t, X_y\}$
    Record the evaluation outcome in P
  End For
End For
For each fold i
  For each F_k, where $k = 1, 2$
    For each M_j, where $j = 1, 2, 3, 4, 5$
      Q = Train M_j using $\{T_x, T_y\}$ with F_k
      Evaluate M_j using $\{X_t, X_y\}$
      Record the evaluation outcome in P
    End For
  End For
End For
Save the best model Q based on outcomes in P
Input text T into Q
Obtain the predicted label from Q for T
Stop

3. Results and Discussion

Table 3 shows the performance of each model with no feature optimization. Overall, the RF and LR models outperform the others. They both have an average accuracy of 60.70%. The average accuracy of NB, DT, and SVM is 55.27%, 58.61%, and 56.4%, respectively.

Table 4 presents the average performance of each model utilizing PCA.The results of Table 4 show that the effect of PCA decreases the overall performance of the models. Where RF provides the highest outcome of accuracy 53.63%. The optimization by PCA reduces the relevancy among features. Which leads to a poorer outcome than before.

Table 3. Average performance of each model without feature optimization.

Models	Accuracy	Precision	Recall	F1 Score
RF	60.70	77.40	59.97	66.34
NB	55.27	46.41	59.23	50.0
DT	58.61	54.80	59.60	56.28
SVM	56.47	82.19	55.92	64.93
LR	60.70	48.01	64.69	53.56

Table 4. The average performance of each model with PCA.

Models	Accuracy	Precision	Recall	F1 Score
RF	53.63	69.45	52.86	59.53
NB	51.14	98.40	50.57	66.8
DT	51.89	50.95	51.81	51.32
SVM	51.09	95.80	50.54	66.08
LR	53.08	78.00	52.98	62.14

Table. 5 presents the average performance of each model utilizing LDA. The results of Table 5 show that the effect of LDA increases the overall performance of the models. Where NB provides the highest outcome of accuracy 99.75%. LDA optimization enhances the relevance among features, resulting in a superior outcome compared to the previous results.

Table 5. The average performance of each model with LDA.

Models	Accuracy	Precision	Recall	F1 Score
RF	98.76	98.58	99.05	98.81
NB	99.75	100	99.51	99.75
DT	97.26	97.16	97.61	97.38
SVM	98.01	99.53	96.77	98.13
LR	98.26	99.05	97.66	98.35

When NB was applied directly to the raw text features, it achieved an accuracy of only 55.27%. However, after applying LDA for feature extraction, the NB model's performance surged to 99.75%. This result highlights the critical role of LDA in transforming high-dimensional, sparse text data into a lower-dimensional, more discriminative space, allowing the Naive Bayes classifier to achieve near-perfect classification performance.Figure 4 presents the comparison of the best models obtained with and without feature optimization techniques. This comparison shows the performance of the best model due to LDA outperforming all other techniques massively. This best outcome due to LDA is obtained by using the NB model. Because of offering the maximum outcome LDA with NB is utilized as the final model to determine the category between ChatGPT generated and web text.

Table 6 summarizes the best model's (NB+LDA) precision, recall, and F1 score for each class ("ChatGPT" and "Other"). It shows very high performance for both classes, with F1 scores of 97.90 and 99.30 respectively, indicating excellent class-specific accuracy.

In Figure 5, the confusion matrix visualizes the true versus predicted labels of the ultimate model (NB with LDA). The model correctly classified 201 "ChatGPT" samples and 200 "Other" samples, with only 1 misclassification, demonstrating very strong classification ability.

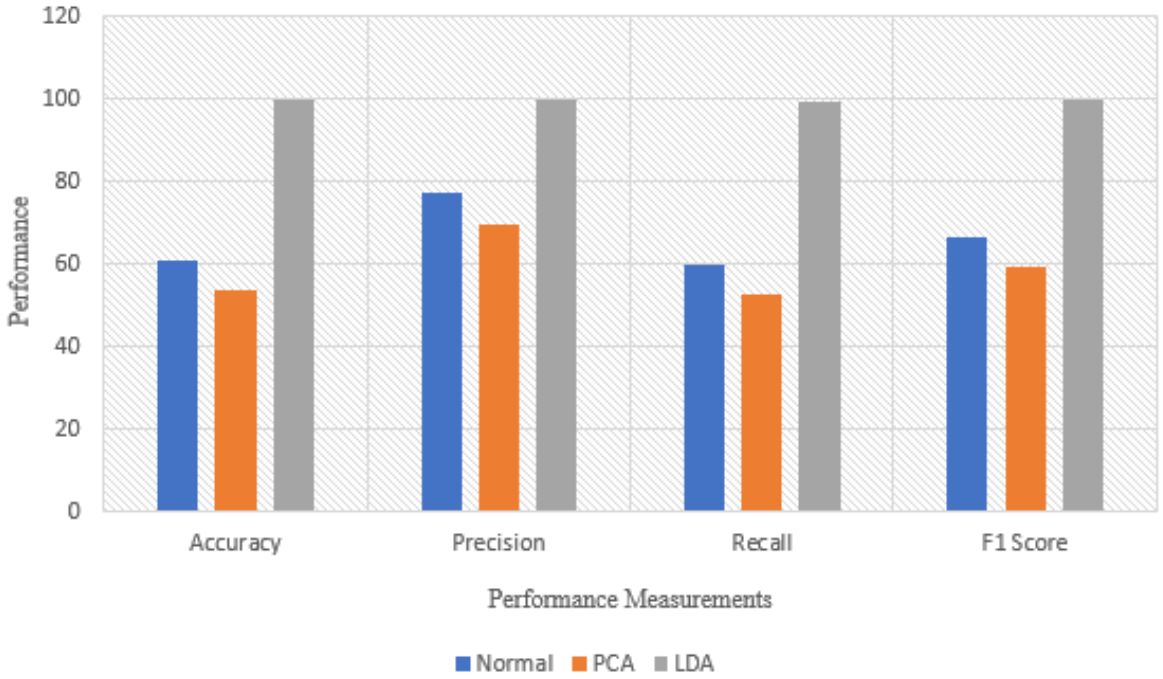

Figure 4. Performance comparison among the best models of different techniques.

Table 6. Class-wise performance of the best model (NB+LDA)

Class	Precision	Recall	F1 Score
ChatGPT	98.00	97.80	97.90
Other	99.50	99.10	99.30

In Figure 6, the ROC curve shows the trade-off between true positive rate and false positive rate of the best model-NB with LDA. A near-perfect curve with an area under the curve (AUC) of 1.00 confirms that the model has outstanding discriminatory power between the two classes.

This research also uses TF-IDF with CV to analyze the effect. Table 7 shows the outcome for this technique. The outcome of this table shows that TF-IDF offers poor performance compared to CV.

Table 7. The average performance of each model with TF-IDF.

Models	Accuracy	Precision	Recall	F1 Score
RF	97.90	98.00	97.80	97.90
NB	99.30	99.50	99.10	99.30
DT	96.50	96.60	96.40	96.50
SVM	97.70	98.10	97.30	97.70
LR	97.80	97.90	97.70	97.80

To observe the effect of non-linear feature optimization techniques besides PCA and LDA, this research uses t-SNE, UMAP [23], with the best performing model NB. Table 8 presents outcomes for these approaches using NB. The outcome demonstrates poor performance compared to linear techniques.

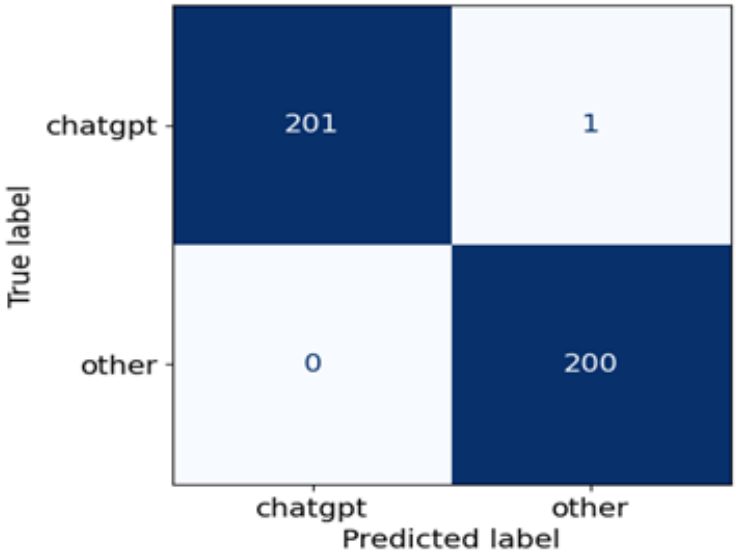

Figure 5. Confusion matrix of ultimate model (NB with LDA)

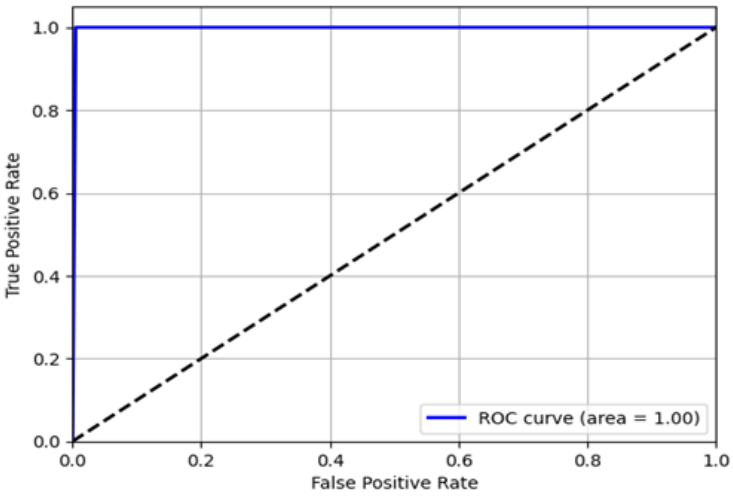

Figure 6. ROC curve of ultimate model (NB with LDA)

Table 8. The average performance of non-linear feature optimization techniques with best classifier.

Techniques	Accuracy	Precision	Recall	F1 Score
t-SNE	98.90	99.00	98.80	98.90
UMAP	98.40	98.80	98.10	98.40

During our analysis of existing research, we have found most of the methods tried to develop ML-based approaches to classify the text between ChatGPT-generated and human-written. Hence, this research also deploys the proposed method on such a dataset that contains the label text of ChatGPT and human written. The dataset is taken from[22]. Table 9, presents the performance of the proposed method on this dataset. The information in this

Table 9. The average performance of different models with LDA for detection of ChatGPT generated and human-written text.

Models	Accuracy	Precision	Recall	F1 Score
RF	86	89.29	86.21	87.72
NB	98.67	97.37	100	98.67
DT	94	96.43	93.10	94.74
SVM	96	96.43	96.43	96.43
LR	96	94.74	97.29	99.59

table shows that our approach successfully classifies the text of ChatGPT and human written with the best accuracy of 98.67% by using the combined model of LDA and NB.

Table 10 shows the comparison between existing works and the proposed method. From the analysis of this table, we get the proposed approach as the first method to distinguish between the text of ChatGPT and web search utilizing Machine learning. The comparison of Table 10 also proves that the proposed method can significantly distinguish not only the text of ChatGPT and web search but also ChatGPT and human writing.

Table 10. Comparison of this research with SOTA methods

Authors	Core method	Text detection	Accuracy
Katib et al. [21]	TF-IDF + word embedding + count vectorizers + TSA-LSTMRNN	ChatGPT-generated and Human written	93.83%
N. Islam et al. [22]	TF-IDF vectorizer + Extremely randomized trees	ChatGPT-generated and Human written	77%
Shijaku and Canhasi [23]	TF-IDF vectorizer + XGBoost	ChatGPT-generated and Human written	96%
Proposed method	LDA + NB	ChatGPT-generated and Human written	98.67%
		ChatGPT-generated and Web Result	99.75%

4. Conclusion

A powerful cutting-edge technology is ChatGPT. We aim to explore whether we can use ML to differentiate between query results from web searches and ChatGPT. In order to complete the investigation, this study uses a balanced dataset with 2010 samples of web search and ChatGPT question results to train five distinct machine learning algorithms. LDA and PCA are the two feature optimization methods that are tested with each of these machine learning models. All of the trial results were analyzed, and it was found that the combination of NB and LDA produced the highest accuracy of 99.75%. Additionally, our combination approach surpasses SOTA approaches in identifying ChatGPT-generated and human-written text from an existing dataset with an accuracy of 98.67 %. This experiment, however, is confined to a dataset capable of categorizing between two classes and a few ML models using only two feature optimization strategies. Which will be expanded upon in the future using multilingual sources of data (e.g., Bengali, Spanish), adversarial prompts (e.g., paraphrased or noisy queries), various advanced ML models, and deep learning techniques like BERT, RoBERTa. We will also try to distinguish between text generated by different AI models (e.g., ChatGPT vs. Gemini) and beyond text classification, the proposed framework will also be extended to detect AI-generated content in other modalities such as images and videos.

Dataset Availability Statement

The dataset of this research can be accessed through the DOI: http://dx.doi.org/10.13140/RG.2.2.21583.80801

Ethical and Practical Concerns

The proposed model could potentially be misused for purposes like automated censorship, surveillance, or unfair discrimination against AI-generated content. To mitigate these risks, implementing regular fairness audits and bias evaluations is crucial to ensure the model treats all content sources equitably. Additionally, transparency in model design and open access to evaluation metrics will help prevent misuse. Considering the rapid evolution of ChatGPT models and dynamic changes in web content, the model's robustness must be continuously evaluated. Future updates will include retraining with newer ChatGPT versions (e.g., GPT-4, GPT-5) and diverse, real-time web data. Regular performance benchmarking against evolving datasets will be prioritized to maintain validity. Furthermore, adversarial testing strategies will be adopted to detect vulnerabilities and adapt the model over time.

REFERENCES

1. Radford, A., et al. "Language models are unsupervised multitask learners." OpenAI Blog, 1(8), 2019.
2. Vaswani, A., et al. "Attention is all you need." Advances in neural information processing systems, 2017.
3. Wang, A., et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding." arXiv preprint arXiv:1804.07461, 2018.
4. Wang, A., et al. "Superglue: A stickier benchmark for general-purpose language understanding systems." arXiv preprint arXiv:1905.00537, 2019.
5. Baeza-Yates, R., & Ribeiro-Neto, B. "Modern information retrieval." Addison-Wesley, 2011.
6. Chakraborty, S., Bedi, A. S., Zhu, S., An, B., Manocha, D., & Huang, F. "On the possibilities of ai-generated text detection." arXiv preprint arXiv:2304.04736, 2023.
7. Harada, A., Bollegala, D., & Chandrasiri, N. P. "Discrimination of human-written and human and machine written sentences using text consistency." In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 41-47). IEEE, 2021.
8. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. "Detectgpt: Zero-shot machine-generated text detection using probability curvature." arXiv preprint arXiv:2301.11305, 2023.
9. Gehrmann, S., Strobelt, H., & Rush, A. M. "Gltr: Statistical detection and visualization of generated text." arXiv preprint arXiv:1906.04043, 2019.
10. Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Ghaffar, M. B., & Khan, W. (2024). Sentiment analysis of movie reviews based on NB approaches using TF–IDF and count vectorizer. Social network analysis and mining, 14(1), 87.
11. Kansal, T., Bahuguna, S., Singh, V., & Choudhury, T. "Customer segmentation using K-means clustering." In 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS) (pp. 135-139). IEEE, 2018.
12. Bro, R., & Smilde, A. K. "Principal component analysis." Analytical methods, 6(9), 2812-2831, 2014.
13. Balakrishnama, S., & Ganapathiraju, A. "Linear discriminant analysis—a brief tutorial." Institute for Signal and information Processing, 18(1998), 1-8.
14. Evgeniou, T., & Pontil, M. "Support vector machines: Theory and applications." In machine learning and Its Applications: Advanced Lectures (pp. 249-257). Springer Berlin Heidelberg, 2001.
15. Breiman, L., & Cutler, A. "Random forests." Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324, 2001.
16. Charbuty, B., & Abdulazeez, A. "Classification based on decision tree algorithm for machine learning." Journal of Applied Science and Technology Trends, 2(01), 20-28, 2021.
17. Yang, F., Zhang, Y., & Yang, C. "A Comparative Study of machine learning Techniques for Spam Email Classification." Journal of Emerging Trends in Computing and Information Sciences, 2013.
18. Kuha, J., & Mills, C. "On group comparisons with logistic regression models." Sociological Methods & Research, 49(2), 498-525, 2020.
19. Hossain, M. M., Swarna, R. A., Mostafiz, R., Shaha, P., Pinky, L. Y., Rahman, M. M., ... & Iqbal, M. S. "Analysis of the performance of feature optimization techniques for the diagnosis of machine learning-based chronic kidney disease." Machine Learning with Applications, 9, 100330, 2022.
20. Katib, I., Assiri, F. Y., Abdushkour, H. A., Hamed, D., & Ragab, M. "Differentiating Chat Generative Pretrained Transformer from Humans: Detecting ChatGPT-Generated Text and Human Text Using Machine Learning." Mathematics, 11(15), 3400, 2023.
21. Islam, N., Sutradhar, D., Noor, H., Raya, J. T., Maisha, M. T., & Farid, D. M. "Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning." arXiv preprint arXiv:2306.01761, 2023.
22. Shijaku, R., & Canhasi, E. "ChatGPT Generated Text Detection." Publisher: Unpublished, 2023.
23. Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2021). Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. Journal of Machine Learning Research, 22(201), 1-73.