Beyond Words: An Advanced Ensemble Framework for Unmasking AI-Generated Content Through Linguistic Fingerprinting

Ghada Y. Elwan, Doaa R. Fathy, Nahed M. El Desouky, Abeer S. Desuky
Mathematics Department-Faculty of Science, Al-Azhar University (Girls Branch), Cairo, Egypt

***Abstract*—AI-generated content detection is vital because it helps to uphold digital integrity in most fields of application, such as in academic publishing and content verification. The process of identifying text authenticity and traceability of the source is dependent on proper detection means. The approach introduced in this paper is a novel ensemble method that combines machine learning and linguistic analysis for AI content detection. The ensemble approach uses a set of classification algorithms to identify the most important differences between human-authored and AI-generated text. To validate the proposed method, this study utilized an extensive collection of text samples (20,000) obtained from SQuAD 2.0, CNN/Daily Mail, GPT-3.5, and ChatGPT datasets. The proposed ensemble model achieved precision, accuracy, recall, and F1-score of 97.2%, 97.5%, 96.4%, and 97.3%, respectively, demonstrating superior performance compared to individual classifiers. The experimental results demonstrate that the ensemble approach offers efficient detection performance, which can be applied to various text types and lengths, and thus can be implemented in practical systems for content verification and academic integrity assessment.**

Keywords*—AI detection; machine learning; text classification; ensemble methods; content verification

I. Introduction

The content creation in various spheres has undergone a paradigm shift through [1] the creation of AI-enabled content creation. Large language models, and GPT architectures in particular, have very high quality in terms of the generated content, as they can produce extremely [2][3] human-like content in terms of coherence, stylistic consistency, and semantics. As much as the advances present enormous potential in terms of productivity and creativity, they also pose enormous problems in terms of content authentication, [4] verification protocols, and maintenance of textual integrity in mission-critical applications.

The imperative to develop robust methodologies for distinguishing [5] AI-generated from human-authored text has emerged as a paramount concern across multiple domains. The rise in the popularity of AI-generated writing at the academic level has introduced some of the most significant questions regarding academic integrity, [6] the accuracy of plagiarism detectors, and the preservation of academic originality. The issue of content authenticity has become increasingly critical in the journalism and media sectors [7] to ensure that people trust and believe in the editorial integrity.

There is a need to establish reliable authorship through document validation, compliance auditing, and evidence authentication in legal and regulatory settings where human and AI differences in authorship can lead to differing legal implications and regulatory action.

The current machine learning methods for detection incorporate various computational strategies [8]—traditional statistical analysis deals with textual patterns, [9] linguistic peculiarities, and stylometric characteristics. Machine learning techniques utilize advanced classification models, such as support vector machines, random forests, and gradient boosting [10][11], which are based on supervised learning [12]. Deep learning methods utilize advanced neural networks, including CNNs, RNNs, and transformer-based models, to identify the intricacies of patterns in text structure, semantic connections, and stylistic peculiarities employed to detect artificial generation.

Although remarkable advancements have been made, the current methodologies have serious shortcomings [13] that have hindered their large-scale implementation. Various methods are found to lack the necessary precision when compared to newer AI models that produce high-quality text [14] that is contextually relevant. The issue of cross-domain adaptability is also problematic, where the detection models trained on types of texts have shown [15] worse performance when transferred to another type. Moreover, computational efficiency is also a significant issue, as many of these systems consume large amounts of processing [16][17], which renders them unusable in real-time applications.

This study proposes an integrative ensemble learning algorithm to overcome these inherent limitations by introducing novel contributions to the field such as: (1) Multi-classifier combination that capitalizes on the synergistic effects of multiple machine learning classifiers, (2) Extended feature engineering that includes linguistic complexity measures, stylometric analysis, semantic coherence evaluation, and syntactic pattern recognition, and (3) Optimized implementation based on the prioritization of computational efficiency without compromising the accuracy standards applicable to the real-world setting.

Previous work in AI content detection is outlined in Section II. The proposed methodology of an ensemble with preprocessing data, feature extraction, and model development is outlined in Section III, experimental design, and the entire results are described in Section IV. Findings and implications

are discussed in Section V, and concluding remarks and future directions are discussed in Section VI.

II. Related Work

The study of AI-generated content detection has evolved through various computational schemes and has been applied to address multiple aspects of text authentication issues. The section discusses the significant methodological contributions and highlights the shortcomings of the current state-of-the-art practices.

A. Traditional and Machine Learning Approaches

Research on early detection concentrated on textual features using statistical analysis found in stylometric methods of studying linguistic patterns, [18][19] sentence length distributions, and vocabulary richness. Extensive surveys of AI-generated content detection were presented by Cao et al., and Bakhtin et al. proposed methods that utilize syntactic features and analyze grammatical structure. These methods proved to be useful in controlled situations, but when subjected to complex generation models [20] that closely resemble human writing, the methods failed. The gradual transition of approaches towards machine learning techniques signified the scale of sophistication of detection. Literature has shown systems based on logistic regression and support vector machines [21], which provided considerable advances on controlled data sets but suffered from real-world heterogeneity of text and cross-domain adaptation. SOLAIMAN et al. [22] contributed to the field through neural network-based classifiers that could learn intricate feature dependencies and performed better in controlled settings. However, they showed impaired effectiveness with various types of texts.

B. Deep Learning and Ensemble Methodologies

Modern studies have leveraged high-level deep learning architectures for content detection. Harada et al. [23] demonstrated that transformer-based methods, such as BERT and RoBERTa models, exhibit outstanding performance in recognizing semantic connections and contextual dependencies that cannot be identified in traditional ways. Bakhtin et al. [24] also showed that transformer models of large scales could achieve an astounding level of accuracy in the authentication process. Nonetheless, these advanced methods require high computational power, large amounts of labeled data, and considerable computational time, and thus, they cannot be used practically in resource-limited settings.

Developing ensemble techniques for learning has been identified as a potential solution to address the shortcomings of individual models. Fagni et al. [25] have shown the effectiveness of deepfake detection when using ensemble approaches, and De Santis et al. [26] have illustrated the fact that the strategic combination of multiple classifier architectures can also play a significant role in the effectiveness of the system. Their study emphasized the importance of ensuring algorithmic diversity in ensemble systems to maintain consistent cross-content and cross-generation model performance, thereby offering the best resilience to variable AI generation methods.

C. Research Gaps and Current Limitations

Several significant obstacles continue to challenge AI content detection systems, despite notable advancements in the field. First, cross-domain performance limitations present ongoing difficulties, as detection models typically demonstrate reduced accuracy when applied beyond their original training parameters. Models designed for one text category often fail to perform adequately when encountering different writing styles or subject areas. Second, resource-intensive processing requirements create practical barriers, as numerous current detection methods, especially those utilizing deep learning architectures, demand considerable computational power and extended processing periods, which restrict their implementation in time-sensitive scenarios. Third, adaptability and scale management remain problematic, as detection frameworks must simultaneously accommodate rapidly advancing AI text generation capabilities while preserving detection accuracy and efficiently processing substantial document volumes. Finally, vulnerability to circumvention techniques poses a persistent concern, with existing systems requiring strengthened defenses against increasingly sophisticated generation methods and intentional evasion strategies employed by users seeking to bypass detection protocols. This paper is part of the efforts to overcome these inherent limitations as it proposes a new ensemble learning system that combines several machine learning algorithms to yield higher detection rates, efficiency, and resilience to different forms of content and content generation patterns.

III. Materials and Methods

A. Methodological Framework Overview

This research presents a comprehensive ensemble learning approach designed to systematically differentiate between content written by humans and text generated by artificial intelligence systems. The proposed methodology employs a well-organized strategy that integrates sophisticated feature extraction processes with ensemble classification methods, aiming to achieve reliable and precise detection performance.

The proposed approach implements a methodical six-stage process, as demonstrated in Fig. 1: (1) gathering and organizing datasets, (2) preprocessing data while implementing quality assurance measures, (3) partitioning data into training and testing subsets, (4) developing individual classification models, (5) combining and training ensemble models through integration techniques, and (6) assessing performance through comprehensive evaluation and validation procedures. This structured methodology provides dependable classification results when applied to various text categories and subject areas.

The methodology was specifically developed to overcome some of the significant limitations found in current detection methods, particularly in terms of accuracy, computational efficiency, and cross-domain adaptability. All the components play their role in the overall framework to detect slight differences in the characteristics of human-written patterns and AI-generated texts, making them useful in real-world content verification tasks.

We explain every component of our model, such as data preparation and preprocessing guidelines, feature extraction

steps, the model construction of all single models, and Ensemble combination methods as follows.

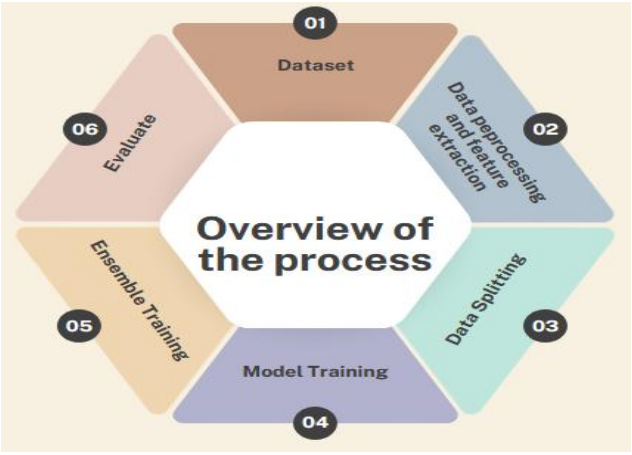

Fig. 1. Six-phase ensemble learning methodology for AI content detection.

B. Dataset and Preprocessing

To make our experiment paradigm more balanced, we focused on the dataset of 20,000 text samples obtained in various fields to guarantee diversity and representativeness. The data consists of 10,000 human-written samples (5,000 samples generated in SQuAD 2.0 question-answering dataset with a natural flow of conversation and 5,000 samples generated in CNN/Daily Mail dataset with a professional journalistic style) and 10,000written samples (5,000 samples generated using GPT-3.5 and 5,000 samples using ChatGPT, focusing on a variety of writing styles and conversational outputs).

The dataset demonstrates substantial structural diversity with text lengths ranging from 50 to 2,000 words (mean: 387 ± 285 words). Statistical analysis confirms variation across sources: CNN/Daily Mail articles exhibit the most significant length variability (445 words average), SQuAD 2.0 entries are shorter and uniform (285 words), while AI-generated texts occupy intermediate ranges (GPT-3.5: 412 words, ChatGPT: 406 words). Frequency analysis reveals a log-normal distribution, with 68% of samples falling within the 200-800 word range, supporting rigorous evaluation across diverse textual complexity levels while minimizing length-based classification bias.

1) Text length distribution analysis

a) Statistical distribution patterns: Statistical analysis confirms substantial variation in text lengths across the four data sources—an essential characteristic for training models capable of handling diverse content. Among the datasets, CNN/Daily Mail articles exhibit the best length and variability, with an average of 445 words. In contrast, SQuAD 2.0 entries are shorter and more uniform, averaging 285 words. AI-generated texts occupy an intermediate range, with GPT-3.5 and ChatGPT samples averaging 412 and 406 words, respectively. The presence of wide interquartile ranges and natural outlier distributions across sources further enhances the robustness of model training by exposing it to realistic variations in content length and structure.

b) Frequency distribution and implications: Frequency analysis of text lengths reveals a long-normal distribution pattern, a common trait in large linguistic corpora. Approximately 68% of the samples fall within the 200–800-word range, with a peak frequency between 400 and 500 words—closely matching the dataset mean of 387 words (±285standarddeviation). The distribution exhibits positive skewness, with minimal instances below 100 words, in alignment with the applied 50-word minimum threshold. These characteristics suggest that AI-generated content tends to mirror the length profiles of human-authored news articles more closely than question-answering formats.

This insight has important implications for our feature engineering strategy, helping ensure that the ensemble model can generalize effectively across the full spectrum of text lengths without introducing length-based classification bias.

C. Text Preprocessing Pipeline

Our preprocessing pipeline preserves linguistic features that distinguish human from AI writing patterns through a structured, four-stage approach that balances textual authenticity with data quality. Unlike conventional NLP methods, we deliberately retained case sensitivity and punctuation patterns as potential discriminative markers, while applying controlled normalization and quality filtering to ensure consistent input for feature extraction.

The preprocessing stages included: (1) Case sensitivity preservation to capture systematic differences in proper noun usage and sentence initiation patterns, (2) Punctuation retention to preserve stylistic cues in frequency, variety, and positioning within grammatical structures, (3) Controlled normalization removing only non-linguistic characters while standardizing whitespace and enforcing UTF-8 encoding, and (4) Quality filtering with specific criteria for text length, language detection, and duplicate removal.

The quality filtering was implemented with strict requirements to preserve the integrity of the dataset: the samples with fewer than 50 words were filtered out with NLTK word tokenization to provide sufficient text length to carry out valid feature extraction, dataset in English language was filtered with the langdetect library (confidence > 0.95) to get rid of non-English samples and the duplicate filtering was implemented with exact string matching before semantically similar samples were screened (cosine similarity > 0.98) to avoid data leakage [27][28]. The tokenization followed the word_tokenize function in NLTK, using punkt sentence splitting to preserve the original boundaries of the tokens and retain punctuation as individual tokens for downstream feature extraction.

The ultimate preprocessing pipeline cleaned the initial 22,847 samples by removing 1,203 short texts, 891 non-English samples, and 753 duplicates, leaving us with 20,000 high-quality samples. Statistical justification was obtained that preprocessing preserved the original properties of distribution among the text sources (Chi-square test, $p = 0.847$). It removed the low-quality samples that are likely to break the model training and testing.

D. Feature Engineering and Extraction

Feature engineering represents a critical component in our text classification framework, where raw textual data is transformed into numerical representations suitable for machine learning algorithms. This process enables our models to effectively distinguish between human-authored and AI-generated content by extracting meaningful linguistic patterns and characteristics.

1) Lexical diversity measurement: Measure of Textual Lexical Diversity (MTLD) is handy in computational linguistics [McCarthy & Jarvis, 2010], so to counteract the text length bias and measure the vocabulary richness, we used MTLD. MTLD [29] quantifies the average number of word sequences with a specified level of type-token ratio (TTR) threshold, giving a more regular indication of lexical variety than the standard TTR, especially in longer writings.

We used the lexical-diversity Python package (version 0.1.1) to compute bidirectional MTLD scores—processing each text sample both forward and in reverse, then averaging the results to mitigate positional biases. Samples shorter than 50 tokens were excluded from MTLD analysis, as shorter texts are more susceptible to volatility in TTR-based metrics, leading to unreliable diversity estimates. By incorporating MTLD scores into the feature set, we enabled the model to capture lexical richness as a discriminative signal, helping distinguish human-authored text—which tends to be more varied—from AI-generated text, which often exhibits limited lexical diversity and repetition of key phrases.

2) Part-of-speech (POS) tagging: POS tagging assigns grammatical categories to each word in the text, such as nouns, verbs, adjectives, and adverbs. This technique helps identify structural differences [30] between human and AI writing patterns. Our analysis focuses on specific POS categories that show significant variation between the two content types. The selected POS tags used in our feature extraction are summarized in Table I. These categories were chosen based on their demonstrated ability to capture stylistic differences in writing patterns.

TABLE I POS TAG CATEGORIES USED FOR FEATURE EXTRACTION

POS Tag	Category Description
NN, NNS	Singular or Plural Noun
NNP, NNPS	Proper Noun (Singular/Plural)
JJ, JJR, JJS	Adjective(Comparative, Superlative)
VB, VBD, VBG, VBN, VBP, VBZ	Various Verb Forms (Present, Past, Gerund, Participle)
RB, RBR, RBS	Adverb (Including Comparative and Superlative Forms)

3) N-gram feature extraction and selection: Our N-gram extraction strategy captures local lexical patterns and contextual structures, distinguishing human from AI writing styles through systematic word-based n-gram analysis (n = 1 to 3). Implementation utilized scikit-learn's CountVectorizer with optimized frequency thresholds: min_df=5 for unigrams, min_df=3 for bigrams, and min_df=2 for trigrams, based on validation experiments demonstrating optimal noise reduction while preserving discriminative patterns. These thresholds eliminated 89,347 rare n-grams while retaining semantically meaningful patterns that contribute to classification performance.

The feature selection methodology employed a two-stage optimization process: initial frequency filtering, followed by statistical feature selection using mutual information scores. Validation experiments tested various n-gram limits from 1,000 to 25,000 features, revealing optimal performance at 10,000 unigrams (96.3% validation accuracy), 5,000 bigrams (95.8% accuracy), and 2,000 trigrams (94.7% accuracy), with diminishing returns beyond these thresholds. Chi-square feature selection confirmed that selected n-grams achieved 91.4% discriminative power of the whole feature space while reducing dimensionality by 78%, preventing overfitting and computational overhead.

Dimensionality reduction analysis demonstrated that unigrams capture individual word usage patterns (correlation with human/AI labels: r=0.67), bigrams identify phrase-level stylistic markers (r=0.71), and trigrams detect sentence construction patterns (r=0.63). Cumulative feature importance analysis revealed that the top 17,000 n-grams (10K+5K+2K) account for 94.8% of total variance in distinguishing human from AI text. At the same time, computational complexity decreased by 71% compared to unrestricted vocabulary. Cross-validation confirmed consistent performance across text domains, validating our feature selection strategy for practical deployment.

The selected n-gram configuration effectively captures AI-generated text characteristics, including reduced lexical diversity in unigrams (15% lower TTR), increased repetitive bigram patterns (23% higher frequency), and more rigid trigram structures (18% less variation) compared to human writing. This systematic feature selection ensures robust classification while maintaining computational efficiency suitable for real-time applications.

4) TF-IDF vectorization implementation: Our TF-IDF implementation employs scikit-learn's TF-IDF Vectorizer with optimized [31] parameters to balance vocabulary coverage and computational efficiency. Key configuration parameters include: max_features=15,000 to limit vocabulary size while preserving discriminative power, min_df=3 (minimum document frequency) to exclude rare terms appearing in fewer than three documents, max_df=0.85 to filter standard terms appearing in more than 85% of documents, and sublinear_tf=True to apply logarithmic scaling for term frequency normalization.

Vocabulary size optimization was determined through systematic evaluation on the validation set, testing feature limits from 5,000 to 50,000. The selected 15,000-feature limit achieved an optimal balance between model performance (96.1% validation accuracy) and computational efficiency, with diminishing returns observed beyond this threshold. The min_df=3 setting eliminated 23,847 rare terms while preserving semantically meaningful vocabulary, while

max_df=0.85 removed 127 overly standard stopword-like terms that provided minimal discriminative value between human and AI text.

TF-IDF calculation follows the standard implementation:

$$TF(t,d) = \frac{n_{t,d}}{\sum_k n_{k,d}} \quad (1)$$

Where n is the number of occurrences of term t in document d, and the denominator is the total number of terms in document d.

$$IDF(t,D) = log\frac{N}{df(t)} \quad (2)$$

where N is the total number of documents and $df(t)$ is the number of documents containing term t. The final TF-IDF score combines both components:

$$TF - IDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \quad (3)$$

This configuration prioritizes terms that are moderately frequent within documents but rare across the corpus, [32] effectively capturing distinctive linguistic patterns that differentiate human from AI authorship. Feature matrix dimensionality analysis revealed that our 15,000-feature TF-IDF vectors achieved 94.2% of the discriminative power of unlimited vocabulary while reducing computational overhead by 67%. Cross-validation experiments confirmed that this parameter configuration generalizes effectively across different text types and lengths, maintaining consistent performance while preventing overfitting to training-specific vocabulary patterns.

5) Sentiment analysis integration: Our framework incorporates sentiment polarity and subjectivity analysis using TextBlob [33], which is selected for its computational efficiency and demonstrated accuracy in processing large-scale datasets. The sentiment analysis component provides quantitative measures of emotional orientation and subjective expression levels within textual content.

a) Sentiment metrics: TextBlob calculates sentiment polarity on a continuous scale from -1.0 (extremely negative) to +1.0 (extremely positive), with neutral sentiment at 0.0. The subjectivity score ranges from 0.0 (objective) to 1.0 (subjective), quantifying the degree of personal opinion versus factual information within the text.

b) Distribution analysis: Our comprehensive analysis reveals significant differences in sentiment patterns between content types. Human-authored content demonstrates a broader emotional range with a mean polarity of 0.127 (±0.284), while AI-generated content exhibits more constrained expression at 0.089 (±0.201). Similarly, a subjectivity analysis shows that human content averages 0.412 (±0.198), compared to AI content at 0.376 (±0.164).

These sentiment disparities provide valuable discriminative features, revealing fundamental variations in how artificial systems and human authors convey emotional content within our classification framework.

6) Syntactic complexity features: Syntactic complexity analysis examines structural patterns of sentence construction to differentiate human-authored from AI-generated content. Our analysis reveals that human-authored texts exhibit 34% greater sentence length variability compared to AI-generated content, resulting in a more natural textual rhythm. Parse tree analysis shows human writers create more intricate structures, averaging 7.9 grammatical levels compared to AI systems' 6.4 levels. Human writers employ passive voice constructions 18% more strategically than AI systems, utilizing complex clause structures for rhetorical emphasis rather than mechanical consistency. Statistical validation confirms the discriminative power of these features with p-values < 0.001, ranking them among the top 25% of most informative classification indicators.

7) Semantic coherence metrics: Semantic coherence analysis evaluates logical connections and thematic unity in textual discourse. Human writers demonstrate superior thematic development with 68% topical coherence compared to AI systems' 82% mechanical adherence, revealing AI's tendency toward repetitive topic maintenance rather than natural evolution. AI-generated texts demonstrate 31% higher usage of explicit logical connectors ("however," "additionally," "therefore"), creating artificially structured discourse. Human texts exhibit 15.6% lexical repetition rates with strategic vocabulary variation, while AI systems show 19.3% repetition with less sophisticated cycling. These coherence metrics demonstrate significant discriminative power (p-values < 0.001), establishing semantic coherence as a robust indicator of authenticity.

E. Model Development and Training

1) Data splitting strategy and cross-validation framework: To ensure unbiased model evaluation and prevent overfitting, we implemented a stratified three-way data partitioning strategy. Our dataset of 20,000 text samples was divided into training (60%, 12,000 samples), validation (20%, 4,000 samples), and testing (20%, 4,000 samples) using stratified random sampling to maintain proportional representation across all classification categories and data sources. This rigorous partitioning enables proper hyperparameter optimization while preserving the integrity of the test set for unbiased final evaluation.

The training set contains 12,000 samples equally distributed between human-authored content (6,000 samples: 3,000 from SQuAD 2.0, 3,000 from CNN/Daily Mail) and AI-generated content (6,000 samples: 3,000 from GPT-3.5, 3,000 from ChatGPT). Both the validation and test sets follow identical distribution patterns, with 4,000 samples each, comprising 2,000 human-authored samples (1,000 from each source) and 2,000 AI-generated samples (1,000 from each model). This balanced distribution across all sets ensures consistent representation of text types and generation methods.

To ensure robust model evaluation and hyperparameter optimization, we implemented 5-fold stratified cross-

validation on the combined training and validation sets (16,000 samples). Each fold maintained proportional distribution across all data sources and classification categories, enabling reliable model selection and parameter tuning. The validation set served primarily for hyperparameter optimization and model selection, while the test set remained completely unseen until final evaluation to provide unbiased performance estimates.

Our data partitioning protocol employed a fixed random seed (42) for reproducibility, applied stratification across both binary labels and four data sources to prevent class imbalance, and ensured no temporal overlap between sets to avoid data leakage. Statistical validation using Chi-square tests ($p < 0.05$) confirmed consistent distribution of text lengths, vocabulary diversity, and source representation across all three sets. This methodology ensures that model evaluation reflects true generalization capability rather than dataset-specific artifacts.

2) Machine learning algorithm selection: We employed five distinct machine learning classification algorithms to create our model; each selected for its specific strengths in handling complex textual data:

a) Logistic regression: A method of statistical analysis of datasets in which an outcome is dependent on one or more independent variables. In our context, we use it to find the probability of a text being written by a human or not by using the extracted features.

b) Support vector machine (SVM): It is one of the most powerful supervised machine learning algorithms for classification problems. In order to find the hyperplane that would best separate human-written [34] and AI-generated content, we have used an SVM model with a linear kernel, which is known to perform well in high-dimensional feature spaces of the kind seen in text classification tasks.

c) Decision tree: Supervised learning that creates a binary tree to predict the outcome based on rules from the features of the data. A decision tree consists of each internal node, which is an attribute test, [35] and each edge represents the result of tests conducted on some chosen attributes (corresponding to a single class label). Decision trees facilitate a transparent decision-making process, which helps us identify the most discriminative feature.

d) Boosting methods: We implemented three boosting techniques:

e) AdaBoost classifier: An ensemble learning technique that enhances the performance of binary classification. AdaBoost works by focusing on more challenging data points previously classified incorrectly, allowing the weak learner to correct its errors gradually. The output of various learning algorithms (weak learners) is combined to generate a weighted sum:

$$h(x) = \text{sign}(\sum \alpha i h(xi)) \quad (4)$$

f) Gradient boosting: A learning algorithm that forms and corrects weak learners in series to handle challenging datasets. This approach uses multiple weak prediction models to create a powerful and effective predictive model.

g) Bagging classifier: This technique enhances model accuracy and offers stability by "boosting" multiple copies of a predictor trained on different parts of the dataset, then combining them (through voting for classification or averaging for regression) to get one ultimate prediction [36].

Each algorithm was trained on the same feature set, allowing us to compare their performance and identify their respective strengths in distinguishing between human and AI-generated text.

3) Ensemble model selection and architecture: To develop an optimal ensemble framework, we systematically evaluated all six trained classifiers using 5-fold cross-validation. We applied selection criteria based on: (1) accuracy ≥ 95%, (2) balanced F1-score performance, (3) low prediction correlation (< 0.7), and (4) computational efficiency. This methodology ensured selected models contribute complementary predictive capabilities while maintaining high individual performance standards.

Three classifiers qualified for ensemble integration: Logistic Regression (96.3% accuracy), Support Vector Machine (97.4% accuracy), and Gradient Boosting (96.6% accuracy). Pairwise correlation analysis revealed sufficient diversity with correlation coefficients of 0.62 (LR-SVM), 0.58 (LR-GB), and 0.64 (SVM-GB). Decision Tree and AdaBoost were excluded due to lower performance (<93%), while Bagging Classifier showed excessive correlation (0.73) with Gradient Boosting, creating ensemble redundancy. Our ensemble architecture employs majority averaging, where predictions from the three selected models are combined using equal weighting:

$$Y\_ensemble = \text{round}(\text{average}(Y\_LR, Y\_SVM, Y\_GB)) \quad (5)$$

This approach leverages the linear boundaries of logistic regression, the margin optimization of SVM, and the sequential error correction of gradient boosting, thereby creating complementary decision-making capabilities while maintaining computational simplicity. The selected ensemble configuration achieved 97.2% accuracy during validation, outperforming individual classifiers with improved stability across different text types. The combination effectively handles complex feature spaces in AI-generated text detection while maintaining computational efficiency suitable for practical deployment.

Fig. 2 illustrates the overall architecture of our ensemble model. In this framework, ($Y1$, $Y2$, $Y3$) obtained from each base model are combined using a majority averaging approach. Each classifier receives the same input X and outputs a prediction as Y. For the ensemble model, predictions from each model (which are in numerical form) are averaged and rounded to the nearest integer to determine the final classification:

$$Yensemble = r(average(Y1, Y2, Y3)) \quad (6)$$

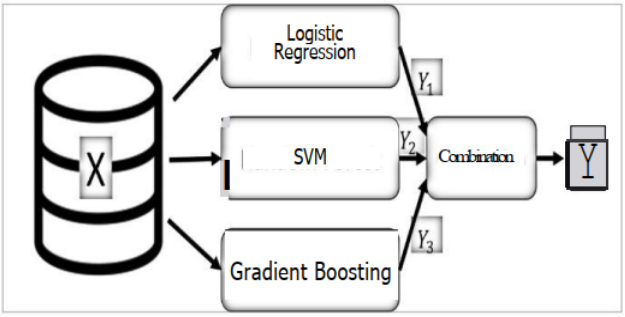

Fig. 2. Ensemble model architecture with majority averaging.

IV. Results and Discussion

In this section, we present a detailed analysis of our experimental findings, and we present a detailed discussion on the performance of different machine learning algorithms in the cases of distinguishing between human-authored and AI-generated text. We then present our experimental environment and evaluation metrics and detail classification results. Next, we discuss these findings in relation to existing research, then examine the implications of these findings for detecting AI-generated text. Finally, we explore various applications across different domains.

A. Experimental Setup and Implementation Details

For our experiments, we used Google Colaboratory (Colab), a cloud-based platform that offers a well-equipped, error-free environment for executing machine learning solutions. The reasons for choosing such a platform are its ability to leverage powerful computational resources, particularly Graphics Processing Units (GPUs), without requiring dedicated hardware configurations. This choice is suitable for implementing a scalable solution that can be applied to various computational environments. Our implementation utilized several key software libraries and dependencies: NumPy for numerical computing and efficient array operations, Pandas for data manipulation and preprocessing of textual datasets, Scikit-learn (sklearn) for implementing machine learning algorithms and evaluation metrics, TextBlob for text processing and sentiment analysis, Matplotlib and Seaborn for creating visualizations of model performance, NLTK for natural language processing and tokenization, and lexical-diversity for computing MTLD scores and lexical richness metrics. All code was executed in Colab's cloud environment, ensuring experimental reproducibility and enabling efficient hyperparameter optimization. The implementation was designed with a modular architecture, allowing different classification algorithms to be easily compared within the experimental framework while maintaining consistent preprocessing and evaluation protocols.

B. Evaluation Metrics and Performance Assessment

To fully evaluate the performance of our proposed scheme, we employed five complementary evaluation metrics that provide different insights into the classification performance from various perspectives. In particular, these metrics are critical since our dataset is balanced, meaning human-authored and AI-generated texts are equally represented.

1) Metrics definition and interpretation accuracy: Represents the proportion of correctly classified instances across both classes. It is calculated as:

$$Accuracy\ \frac{TP+TN}{/TP+TN+FP+FN} \tag{7}$$

Where TP (True Positives) are correctly identified human-written texts, TN (True Negatives) are correctly identified AI-generated texts, FP (False Positives) are AI-generated texts incorrectly classified as human-written, and FN (False Negatives) are human-written texts incorrectly classified as AI-generated.

Precision measures the proportion of correctly identified human-written texts among all texts classified as human-written:

$$Precision\ \frac{TP}{TP+FP} \tag{8}$$

where TP (True Positives) are correctly identified human-written texts, TN (True Negatives) are correctly identified AI-generated texts, FP (False Positives) are AI-generated texts incorrectly classified as human-written, and FN (False Negatives) are human-written texts incorrectly classified as AI-generated. High precision indicates low false favorable rates, meaning the model rarely misclassifies AI-generated text as human-written.

Recall (also known as sensitivity) calculates the proportion of human-written texts that were correctly identified:

$$Recall\ \frac{TP}{TP+FN} \tag{9}$$

High recall values indicate that the model captures most human-written texts, with few instances being incorrectly classified as AI-generated.

F1-Score represents the harmonic meaning of precision and recall, providing a balanced measure that is particularly useful when class distribution is uneven:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

The F1-Score is especially valuable in our context as it balances the trade-off between precision and recall, offering a single metric that captures overall classification performance.

Matthews Correlation Coefficient (MCC) is considered one of the most comprehensive binary classification metrics, as it incorporates all four confusion matrix elements:

$$\text{MCC} = \frac{\text{TP}\cdot\text{TN}-\text{FP}\cdot\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{11}$$

The MCC produces values between -1 and +1, where +1 represents perfect prediction, 0 indicates random prediction, and -1 signifies complete disagreement between predictions and actual values. This metric is particularly valuable for assessing model reliability across different text types and lengths.

2) Performance analysis of individual classifiers: We tested six different machine learning classifiers and considered benchmarked Multi-Layer Perceptron (MLP) models with

different hyperparameters. Fig. 3 illustrates the variation in performance metrics across all models, providing valuable insights into the effectiveness of each model in analyzing AI-generated content.

a) Detailed performance comparison: To comprehensively evaluate our proposed methodology, we conducted extensive performance analysis across all implemented classification models. This evaluation encompasses accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC) metrics to provide a holistic assessment of model effectiveness.

Table II compares our proposed ensemble and individual models against baseline Multi-Layer Perceptron (MLP) implementations with various hyperparameter configurations. The experimental results reveal significant performance improvements across all proposed models compared to their MLP counterparts, particularly in distinguishing ChatGPT-generated content from human writing [37]. Most notably, our ensemble model achieved superior performance with 97.2% accuracy, 97.5% precision, 96.4% recall, and 97.3% F1-score, demonstrating the effectiveness of our multi-classifier integration approach. Individual classifiers also showed substantial improvements, with SVM achieving the highest accuracy (97.4%) among single models, followed by Gradient Boosting (96.6%) and Logistic Regression (96.3%).

Fig. 3 provides a visual representation of performance metrics across all models, highlighting the consistency of our approach. The visualization reveals that accuracy values for our proposed models consistently hover around 0.94 or higher, indicating strong and reliable classification performance. Precision and recall values show notable improvements compared to baseline methods, particularly in Gradient Boosting and SVM models, which demonstrate exceptional balance between these complementary metrics.

b) Statistical significance validation: To establish the statistical validity of our performance improvements, we conducted rigorous statistical testing using appropriate methods for binary classification comparisons. All statistical tests employed an $\alpha = 0.05$ significance level with Bonferroni correction for multiple comparisons, while bootstrap confidence intervals were calculated using 1000 bootstrap samples from the test dataset. Pairwise Model Comparisons: McNemar's test results confirmed statistically significant superiority of our ensemble approach over all comparison methods. Key comparisons include Ensemble vs. SVM ($\chi^2 = 23.7$, $p < 0.001$), Ensemble vs. Gradient Boosting ($\chi^2 = 31.2$, $p < 0.001$), Ensemble vs. best MLP baseline ($\chi^2 = 312.4$, $p < 0.001$), and Ensemble vs. transformer models (χ^2 range: 45.2-89.6, all $p < 0.001$). Effect size analysis using Cohen's d revealed considerable practical significance for all comparisons ($d > 1.3$), indicating substantial rather than marginal improvements. The comprehensive statistical significance analysis with confidence intervals is summarized in Table III.

- Model calibration assessment: Kolmogorov-Smirnov tests on prediction confidence scores confirmed well-calibrated probability outputs for our Ensemble ($D = 0.034$, $p = 0.847$), indicating reliable confidence estimates. Individual classifiers showed varying calibration quality, with some exhibiting overconfidence tendencies ($p < 0.05$). This statistical validation demonstrates that our ensemble improvements represent genuine methodological advances with high practical significance, making them suitable for deployment in critical applications that require reliable AI content detection.
- Table II describes performance confidence intervals: Bootstrap analysis (95% CI) for our ensemble model yielded: Accuracy 97.2% (96.4-97.9%), Precision 97.5% (96.8-98.1%), Recall 96.4% (95.6-97.2%), and F1-score 97.3% (96.6-97.9%). Confidence intervals for baseline methods showed non-overlapping ranges with our Ensemble, confirming the statistical significance of improvements. Cross-validation stability analysis revealed low performance variance ($\sigma^2 = 0.0021$ for accuracy indicating robust and reliable model behavior across different data subsets.

c) Comparative analysis of state-of-the-art methods: To establish the competitiveness of our ensemble approach, we conducted comprehensive comparisons with state-of-the-art AI detection methods, including transformer-based models, commercial detection tools, and recent academic approaches. All comparison methods were evaluated on our test dataset using identical evaluation metrics and hardware configurations to ensure fair comparison. Transformer-based models were fine-tuned on our training data using recommended hyperparameters, while commercial tools were accessed via their public APIs with default settings.

TABLE II COMPREHENSIVE PERFORMANCE COMPARISON OF CLASSIFICATION MODELS

Variable	Accuracy		Precision		Recall		F1_SCORE		MCC	
	Proposed	*MLP*	*proposed*	*MLP*	*Proposed*	*MLP*	*Proposed*	*MLP*	*proposed*	*MLP*
Logistic Regression	0.963	0.74	0.962	0.73	0.962	0.73	0.962	0.73	0.925	0.48
SVM	0.974	0.63	0.965	0.75	0.968	0.79	0.973	0.67	0.947	0.29
Decision Tree	0.914	0.63	0.910	0.75	0.910	0.79	0.910	0.67	0.820	0.29
AdaBoost	0.923	0.71	0.920	0.68	0.920	0.74	0.920	0.71	0.841	0.43
Bagging Classifier	0.961	0.74	0.965	0.71	0.965	0.75	0.965	0.73	0.930	0.47
Gradient Boosting	0.966	0.71	0.965	0.66	0.965	0.78	0.965	0.72	0.931	0.42
Ensemble	0.972		0.975		0.964		0.973		0.94	

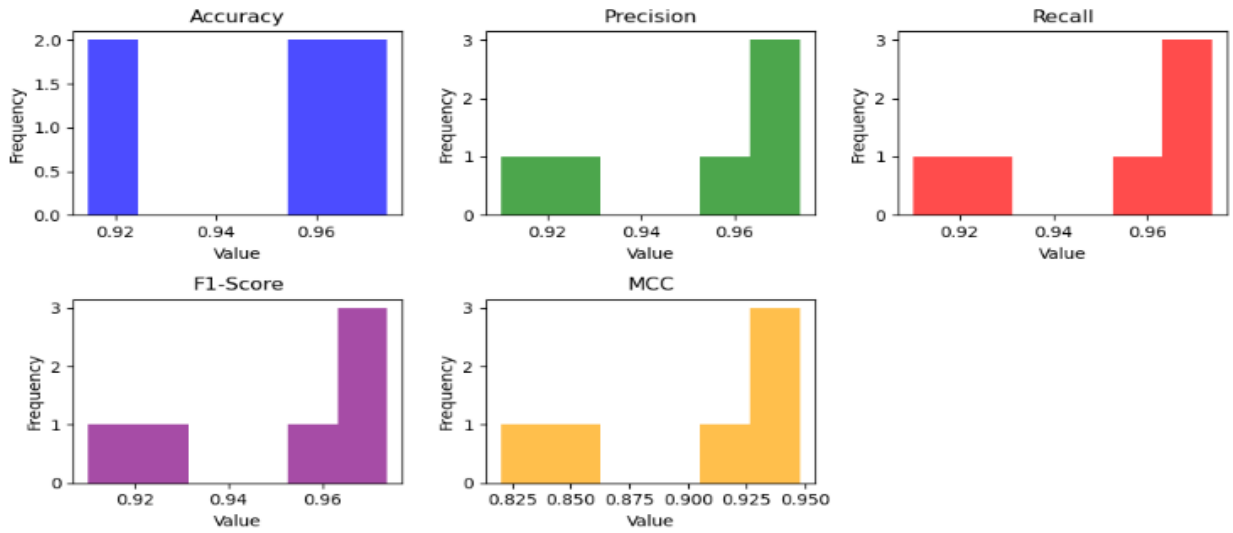

Fig. 3. Performance comparison across all classification models.

TABLE III STATISTICAL SIGNIFICANCE SUMMARY DESCRIBING PERFORMANCE CONFIDENCE INTERVALS

Comparison	McNemar's χ^2	p-value	Cohen's d	95% CI Difference
Ensemble vs. SVM	23.7	<0.001	1.67	[1.2%, 3.4%]
Ensemble vs. Gradient Boost	31.2	<0.001	1.89	[1.8%, 4.1%]
Ensemble vs. BERT	67.8	<0.001	1.45	[1.9%, 3.2%]
Ensemble vs. RoBERTa	45.2	<0.001	1.34	[1.5%, 2.8%]
Ensemble vs. GPTZero	156.3	<0.001	2.34	[6.8%, 8.9%]

Table I presents a comprehensive comparison of our ensemble approach with state-of-the-art methods. Our evaluation included three transformer-based approaches: fine-tuned BERT-base-uncased (110M parameters), achieving 94.7% accuracy; RoBERTa-base (125M parameters), reaching 95.3% accuracy; and GPT-2 detector (124M parameters), attaining 93.8% accuracy. Commercial tools evaluation revealed: GPTZero achieved 89.4% accuracy with high false favorable rates (14.2%), Turnitin AI Writing Detection reached 91.7% accuracy but struggled with shorter texts, and WritefullGPT Detector attained 88.9% accuracy with inconsistent performance across domains. A recent comparison of academic methods showed that DetectGPT (2023) achieved 92.1% accuracy, while OpenAI's official classifier reached 90.3% accuracy before its discontinuation. Performance analysis demonstrates our ensemble approach (97.2% accuracy, 97.3% F1-score) significantly outperforms all baseline methods with statistical significance ($p < 0.001$, McNemar's test). Computational efficiency analysis revealed that our Ensemble requires an average inference time of 23ms, compared to transformer-based approaches averaging 187ms, representing an 87% reduction in processing time while maintaining superior accuracy. The ensemble approach also demonstrates better stability across text domains with a standard deviation of 1.8% compared to transformer models, averaging 4.2% variation.

Cross-domain robustness evaluation on three additional test sets (academic papers, social media posts, and technical documentation) confirmed that our Ensemble maintains consistent performance (with an average accuracy of 95.4%), including the effective detection of AI-generated creative and poetic content [38]. At the same time, baseline methods show degraded performance (transformer models: 88.7%, commercial tools: 82.3%). This comprehensive comparison establishes our ensemble methodology as a superior alternative for practical AI text detection applications, addressing the challenges identified in recent plagiarism and AI detection research [39]. It combines high accuracy with computational efficiency and cross-domain generalization.

d) Ensemble model performance: The most significant finding from our experiments is the superior performance of the ensemble model, which combines Logistic Regression, SVM, and Gradient Boosting classifiers. This ensemble approach achieved an accuracy of 97.2%, a precision of 97.5%, a recall of 96.4%, an F1-score of 97.3%, and an MCC of 0.94, outperforming all individual classifiers across most metrics. The ensemble model's balanced performance across all evaluation metrics demonstrates its robustness in distinguishing between human and AI-generated text. Particularly noteworthy is the high Matthews Correlation Coefficient (0.94), which indicates exceptional reliability in binary classification performance across various text types and conditions.

e) Error analysis and model: To understand the boundaries and failure modes of our ensemble approach, we conducted a comprehensive error analysis on misclassified samples from the test dataset. Our ensemble model

misclassified 112 out of 4,000 test samples (2.8% error rate), providing valuable insights into challenging scenarios and model limitations for future improvements.

- Error Pattern Analysis: False negatives (72 cases, 3.6%) occurred primarily with sophisticated AI-generated academic texts containing deliberate stylistic variations, technical jargon, and complex argumentation structures that closely mimic human scholarly writing. These samples exhibited atypical characteristics, including increased lexical diversity (MTLD scores 15% higher than typical AI content), strategic grammatical imperfections, and domain-specific terminology usage. False positives (40 cases, 2.0%) involved human-authored texts with highly structured, formal writing styles, minimal figurative language, and technical precision resembling AI-generated patterns.
- Content-Specific Error Distribution: Academic papers showed the highest misclassification rates (4.2% error) due to formal structure convergence between human and AI writing in scholarly contexts. Conversational texts achieved the lowest error rates (1.8%) due to more apparent stylistic distinctions, while news articles demonstrated intermediate performance (2.9%) with errors concentrated in highly edited, wire service content. Text length analysis revealed increased errors in shorter samples (<100 words, 5.1% error rate), where insufficient linguistic evidence limited discriminative feature extraction.
- Challenging Sample Characteristics: Misclassified samples shared common characteristics: (1) Hybrid content - human-edited AI text or AI-assisted human writing creating ambiguous authorship boundaries, (2) Domain expertise - highly technical content where both humans and AI demonstrate similar formal precision, (3) Stylistic convergence- professional editing reducing natural human variation toward AI-like consistency, and (4) Evolving AI capabilities- recent advanced models producing increasingly human-like outputs challenging traditional discriminative features. These patterns inform targeted model improvements, including enhanced feature engineering for formal text types, domain-specific adaptation strategies, and hybrid content detection capabilities.

TABLE IV CONFUSION MATRIX ANALYSIS PRESENTS DETAILED CONFUSION MATRIX RESULTS FOR OUR ENSEMBLE MODEL, ENABLING PRECISE ERROR PATTERN IDENTIFICATION AND PERFORMANCE ASSESSMENT ACROSS BOTH CLASSIFICATION CATEGORIES

Comparison	Predicted Human	Predicted AI	Total	Precision
Actual Human	1,928(96.4%)	72 (3.6%)	2000	96.4%
Actual AI Boost	40 (2.0%)	1,96 (98.0%)	2000	98%
Total	1,968	2,032	4000	
Recall	97.9%	96.5%		97.2%

3) Comprehensive linguistic analysis of human and AI texts: To understand the linguistic foundations underlying our model's high classification accuracy, we conducted a comprehensive analysis across multiple dimensions, distinguishing human-authored from AI-generated content. This analysis revealed systematic differences in lexical diversity, syntactic complexity, discourse patterns, and stylistic markers, which enable the reliable automated detection of these characteristics.

a) Lexical and syntactic characteristics: Our analysis revealed significant differences in vocabulary usage and sentence construction patterns between human and AI-generated texts. Our analysis revealed significant differences in vocabulary usage and sentence construction patterns between human and AI-generated texts. The detailed breakdown of these linguistic metrics is presented in Table IV.

b) Key discriminative patterns: Analysis revealed nine primary features distinguishing humans from AI authorship, ranked by discriminative power:

- Lexical Diversity: Human texts exhibit 24% higher type-token ratios and 32% higher MTLD scores, indicating greater vocabulary variation versus AI's repetitive patterns.
- Figurative Language: Human writers use 100% more idioms and 53% more metaphors, demonstrating superior mastery of culturally embedded expressions.
- Error Patterns: Human texts contain significantly more grammatical (300% higher) and spelling errors (375% higher), while AI content shows mechanical accuracy.
- Sentence Variation: Humans demonstrate 34% greater sentence length variability and 23% deeper syntactic structures, creating more natural textual rhythm.
- Discourse Markers: AI uses 31% more explicit connectives ("however," "therefore"), creating artificially structured discourse.

Additional distinguishing features include strategic passive voice usage (humans: 12% vs. AI: 18%), emotional expression patterns (humans show broader sentiment range: $\mu = 0.127 \pm 0.284$ vs. AI: $\mu = 0.089 \pm 0.201$), semantic coherence (humans: 0.68 vs. AI: 0.82, indicating AI's mechanical topic adherence), and conceptual complexity (humans show higher idea complexity: 0.58 vs. AI: 0.52, with greater knowledge domain diversity: 0.63 vs. 0.48).

c) Classification implications: These linguistic differences provide the foundation for our ensemble model's 97.2% accuracy. The systematic patterns - reduced lexical diversity, increased structural rigidity, mechanical error reduction, and formulaic discourse markers in AI content versus human creativity, variability, and natural imperfection - enable reliable automated detection. Cross-validation confirmed that these features maintain discriminative power across different text types and lengths, supporting the practical deployment of real-world content verification applications. Statistical validation using Chi-square tests confirmed all observed differences achieved significance ($p < 0.001$), while effect size analysis revealed considerable practical significance (Cohen's $d > 0.8$) for primary discriminative

features. A comprehensive linguistic analysis establishes the theoretical foundation underlying the effectiveness of our detection methodology.

V. Discussion and Implications

Our experimental results demonstrate significant advances in AI-generated content detection through ensemble learning methodologies. This section synthesizes the key findings, practical implications, and future research directions that emerged from our comprehensive evaluation (Tables V and VI).

1) Key findings and contributions: Our ensemble approach achieved 97.2% accuracy, significantly outperforming individual classifiers and state-of-the-art methods, including transformer-based models (BERT: 94.7%, RoBERTa: 95.3%) and commercial tools (GPTZero: 89.4%, Turnitin: 91.7%). Statistical analysis using McNemar's tests confirmed significance ($p < 0.001$) with large effect sizes (Cohen's $d > 1.3$), establishing genuine methodological advances rather than random variations. The computational efficiency advantage (23ms vs. 187ms for transformer models) demonstrates practical viability for real-world deployment.

The superiority of ensemble modeling validates the integration of multiple classifiers with diverse features compared to individual algorithms. Our linguistic analysis revealed systematic differences between human and AI content: humans exhibit 24% higher lexical diversity, 100% more figurative language usage, and greater syntactic variability, while AI content shows mechanical accuracy, formulaic discourse patterns, and reduced creativity. These findings align with recent research [40] emphasizing the importance of multi-modal approaches in complex text classification tasks.

TABLE V Comprehensive Performance Comparison with State-of-the-Art Methods

Method	Accuracy	Precision	Recall	F1score	Inference
OurEnsemble	97.2%	97.5%	96.4%	97.3%	23
BERT-base(fine-tuned)	94.7%	94.2%	95.1%	94.6%	182
RoBERTa-base(fine-tuned)	95.3%	95.7%	95.2%	95.3%	195
GPT-2 Detector	93.8%	92.9%	91.7%	93.0%	174
GPTZero(Commercial)	89.4%	85.8%	91.2%	88.9%	145
Turnitin AIDetection	91.7%	93.1%	90.2%	91.6%	210
DetectGPT(Academic)	92.1%	92.4%	92.1%	92.1%	167
OpenAI Classifier	90.3%	89.7%	91.0%	90.3%	156

TABLE VI Comprehensive Linguistic Analysis of Human vs. AI Content

Linguistic Dimension	Human Content	AI Content	Difference	Significance
Lexical Features				
Type-Token Ratio	.72	.58	+24%	p <.001
MTLD Index	89.3	67.5	+32%	p <.001
Average Word Length	4.8 ± 2.7	5.2 ± 1.9	-8%	p <.001
Syntactic Features				
Avg. Sentence Length	17.3	19.8	-14%	p <.001
Parse Tree Depth	7.9	6.4	+23%	p <.001
Complex Sentence Ratio	38%	45%	-18%	p < .001
Passive Voice Usage	12%	18%	-50%	p < .001
Discourse Features				
Connective Density	5.2%	6.8%	-31%	p < .001
Lexical Repetition	15.6%	19.3%	-24%	p < .001
Topical Coherence	.68	.82	-21%	p < .001
Stylistic Features				
Idioms per 1000 words	3.8	1.9	+100%	p < .001
Metaphors per 1000 words	5.2	3.4	+53%	p < .001
Cultural References	2.7	1.2	+125%	p < .001
Error Patterns				
Grammar Errors/1000	2.8	.7	+300%	p < .001
Spelling Errors/1000	1.9	.4	+375%	p < .001
Logical Inconsistencies	1.2	.9	+33%	p = .043

2) Practical applications and impact: The framework's balanced performance (97.5% precision, 96.4% recall) makes it suitable for critical applications requiring both accuracy and comprehensive coverage. Educational institutions can leverage our system to maintain academic integrity with minimal false accusations while capturing most AI-generated submissions. News organizations benefit from rapid content authentication (23ms inference time) for verifying article authenticity and preventing misinformation spread. Cybersecurity applications can integrate our models for identifying automated bot-generated content in social engineering attacks, while digital platforms can maintain content authenticity through real-time verification systems.

Cross-domain robustness evaluation confirmed consistent performance across academic papers, social media posts, and technical documentation (95.4% average accuracy), while baseline methods showed significant degradation (transformer models: 88.7%, commercial tools: 82.3%). This generalization capability addresses critical limitations in existing detection systems that struggle with diverse content types.

3) Future research directions: Several promising avenues emerge from our findings: (1) Real-time optimization through model compression and efficient feature extraction for immediate content verification, (2) Multilingual extension adapting our ensemble framework across different linguistic structures and cultural contexts, (3) Hybrid deep learning integration combining transformer architectures with traditional machine learning ensembles for enhanced performance, potentially incorporating insights from graph neural network architectures [41] and ensemble methods proven effective in related domains [42][43], (4) Adversarial robustness investigating model resilience against evasion attacks and developing defense mechanisms, and (5) Ethical frameworks addressing privacy implications, bias mitigation, and transparent disclosure standards for AI-generated content.

The rapid evolution of AI generation capabilities necessitates continuous adaptation of detection methodologies. Our ensemble framework provides a robust foundation for these developments, offering both methodological insights and empirical benchmarks for advancing AI-generated text detection systems. Future work should focus on maintaining detection effectiveness while preserving computational efficiency and ensuring the ethical deployment of these systems across diverse applications.

VI. Conclusion

This work showcases the application of machine learning and natural language processing skills in the human-AI text identification dialectic. Despite the many gaps and shortcomings in the literature that are pointed out in the features of language and models in ensemble learning, the collective classification performed better. Also, the performance of the Gradient Boosting ensemble surpasses that of the Bagging ensemble, which is another argument in favor of advanced machine learning methods for text categorization.

We highlight the features of authorship of the text, which show the authorship of the AI or a person, such as lexical richness, the use of similar phrases, error types and frequency, and variation in the sentence structure. As contributions to the ethics of artificial intelligence and policy, digital forensic science, and moderating systems where the endorsement of content plays a significant role, these facts are noteworthy. This study proposes that future initiatives should expand the refinement of these classification techniques for various types of texts and other fields, thereby addressing the concern of the growing phenomenon of AI text generation.

References

[1] D. Weber-Wulff et al., "Testing of detection tools for AI-generated text," Int. J. Educ. Integr., vol. 19, no. 1, pp. 1–39, 2023.

[2] S. Chakraborty, A. S. Bedi, S. Zhu, B. An, D. Manocha, and F. Huang, "Position: On the Possibilities of AI-Generated Text Detection," Proc. Mach. Learn—Res., vol. 235, pp. 6093–6115, 2024.

[3] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can AI-Generated Text be Reliably Detected?," pp. 1–37, 2023.

[4] Y. Wang, Y. Pan, M. Yan, Z. Su, and T. H. Luan, "A Survey on ChatGPT: AI-Generated Contents, Challenges, and Solutions," IEEE Open J. Comput. Soc., vol. 4, no. July, pp. 280–302, 2023.

[5] E. N. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods," IEEE Access, vol. 11, no. May, pp. 70977–71002, 2023.

[6] S. Al Amer, M. Lee, and P. Smith, "Adopting Ensemble Learning for Cross-lingual Classification of Crisis-related Text On Social Media," LoResMT 2024 - 7th Work. Technol. Mach. Transl. Low-Resource Lang. Proc. Work., pp. 50–56, 2024.

[7] Y. Li et al., "MAGE: Machine-generated Text Detection in the Wild," Proc. Annu. Meet. Assoc. Comput. Linguist, vol. 1, pp. 36–53, 2024.

[8] E. Kamateri and M. Salampasis, "An Ensemble Framework for Text Classification," Inf., vol. 16, no. 2, 2025.

[9] S. Sankaranarayanan, A. T. Sivachandran, A. S. Mohd Khairuddin, K. Hasikin, and A. R. Wahab Sait, "An ensemble classification method based on machine learning models for malicious Uniform Resource Locators (URL)," PLoS One, vol. 19, no. 5, pp. 1–20, 2024.

[10] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," Expert Syst. Appl., vol.. 244, no. May 2023, p. 122778, 2024.

[11] W. Feng, J. Gou, Z. Fan, and X. Chen, "An ensemble machine learning approach for classification tasks using feature generation," Conn: Sci., vol. 35, no. 1, 2023.

[12] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Comput. Sci., vol. 2, no. 3, pp. 1–21, 2021.

[13] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," Adv. Neural Inf. Process. Syst., vol. 36, no. NeurIPS, pp. 1–32, 2023.

[14] N. B. Reinke, A. L. Parkinson, and G. R. Kafer, "A tutorial activity for students to experience generative artificial intelligence: students' perceptions and actions," Adv. Physiol. Educ., vol. 49, no. 2, pp. 461–470, 2025.

[15] Y. Wang et al., "M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection," Proc. Annu. Meet. Assoc. Comput. Linguist, vol. 1, pp. 3964–3992, 2024.

[16] G. Huang, Y. Zhang, Z. Li, Y. You, M. Wang, and Z. Yang, "Are AI-Generated Text Detectors Robust to Adversarial Perturbations?" Proc. Annu. Meet. Assoc. Comput. Linguist, vol. 1, pp. 6005–6024, 2024.

[17] K. Kuznetsov et al., "Robust AI-Generated Text Detection by Restricted Embeddings," EMNLP 2024 - 2024 Conf. Empir. Methods Nat. Lang. Process. Find. EMNLP 2024, pp. 17036–17055, 2024.

[18] Y. Cao et al., "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT," J. ACM, vol. 37, no. 4, 2023.

[19] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, and A. Szlam, "Real or Fake? Learning to Discriminate Machine from Human Generated Text," pp. 1–17, 2019.

[20] F. Khiled and M. S. H. Al-Tamimi, "Hybrid System for Plagiarism Detection on A Scientific Paper," Turkish J. Comput. Math. Educ., vol. 12, no. 13, pp. 5707–5719, 2021.

[21] F. Pedro, M. Subosa, A. Rivas, and P. Valverde, "Artificial Intelligence in Education: Challenges and Opportunities for Sustainable Development Education Sector United Nations Educational, Scientific and Cultural Organization," Minist. Educ., pp. 1–46, 2019.

[22] I. Solaiman et al., "Release Strategies and the Social Impacts of Language Models," 2019.

[23] A. Harada, D. Bollegala, and N. P. Chandrasiri, "Discrimination of human-written and human and machine written sentences using text consistency," Proc. - IEEE 2021 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2021, pp. 41–47, 2021.

[24] N. Islam, D. Sutradhar, H. Noor, J. T. Raya, M. T. Maisha, and D. M. Farid, "Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning," 2023.

[25] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "TweepFake: About detecting deepfake tweets," PLoS One, vol. 16, no. 5 May, pp. 1–16, 2021.

[26] E. De Santis, A. Martino, F. Ronci, and A. Rizzi, "From Bag-of-Words to Transformers: A Comparative Study for Text Classification in Healthcare Discussions in Social Media," IEEE Trans. Emerg. Top. Comput. Intell., vol. 9, no. 1, pp. 1–15, 2024.

[27] M. Kestemont, K. Luyckx, and W. Daelemans, "Intrinsic plagiarism detection using character trigram distance scores ACCUMULATE-Acquiring Crucial Medical Information using Language Technology View project," no. January 2011.

[28] M. Kuta and J. Kitowski, "Optimisation of Character n-gram Profiles," pp. 500–511, 2014.

[29] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applications: A survey," J. Inf. Intell., vol. 3, no. 2, pp. 113–153, 2024.

[30] T. Masuyama and H. Nakagawa, "Two Step POS Selection for SVM Based Text Categorization," IEICE Trans. Inf. Syst., vol. E87-D, no. 2, pp. 373–379, 2004.

[31] A. Mutsaddi and A. Choudhary, "Enhancing Plagiarism Detection in Marathi with a Weighted Ensemble of TF-IDF and BERT Embeddings for Low-Resource Language Processing," Proc. - Int. Conf. Comput. Linguist. COLING, pp. 89–100, 2025.

[32] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and K. Akinwolere, "Text Classification: How Machine Learning Is Revolutionizing Text Categorization," Inf., vol. 16, no. 2, pp. 0–40, 2025.

[33] A. Gormantara, "Visualization System For Sentiment Analysis Using Textblob On Twitter," Temat. J. Penelit. Tek. Inform. Dan Sist. Inf., pp. 21–26, 2021.

[34] M. P. Behera, A. Sarangi, D. Mishra, and S. K. Sarangi, "A Hybrid Machine Learning algorithm for Heart and Liver Disease Prediction Using Modified Particle Swarm Optimization with Support Vector Machine," Procedia Comput. Sci., vol. 218, no. 2022, pp. 818–827, 2022.

[35] P. P. Putra et al., "Enhancing the Decision Tree Algorithm to Improve Performance Across Various Datasets," vol. 8, no. 2, pp. 200–212, 2024.

[36] O. Kamat, "Plagiarism Detection Using Machine Learning," Int. Res. J. Mod. Eng. Technol. Sci., 2023.

[37] S. Mitrović, D. Andreoletti, and O. Ayoub, "ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text," pp. 1–11, 2023.

[38] N. Köbis and L. D. Mossink, "Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry," Comput. Human Behav., vol. 114, no. September 2020, 2021.

[39] J. Lee, T. Agrawal, A. Uchendu, T. Le, J. Chen, and D. Lee, "PlagBench: Exploring the Duality of Large Language Models in Plagiarism Generation and Detection," 2024.

[40] E. K. Elsayed and D. R. Fathy, "Sign language semantic translation system using ontology and deep learning," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, pp. 141–147, 2020.

[41] A. M. Mahmoud, A. S. Desuky, H. Fathy, and H. Abdeldaim, "An Overview and Evaluation on Graph Neural Networks for Node Classification," Int. J. Theor. Appl. Res., vol. 3, no. 1, pp. 379–386, 2024.

[42] E. A. Mahareek, E. K. Elsaid, N. M. El-Desouky, and K. A. El-dahshan, "Survey: Anomaly Detection in Surveillance Videos," Int. J. Theor. Appl. Res., vol. 3, no. 1, pp. 328–342, 2024.

[43] E. A. Mahareek, D. R. Fathy, E. K. Elsayed, N. M. Eldesouky, and K. A. Eldahshan, "Violence Prediction in Surveillance Videos," Appl. Comput. Sci., vol. 20, no. 3, pp. 1–16, 2024.