3rd International Conference on Applications of AI in Smart Technologies and Manufacturing

CERTIFICATE OF PARTICIPATION

This is to certify that Sokalla Aprameya has participated / presented a paper in the 3rd International Conference on Applications of AI in Smart Technologies and Manufacturing (AISTM - 2025) organized by Department of Mechanical Engineering 0n 7 - 8 March, 2025.

Title AI-GENERATED TEXT DETECTION

Convener

Director

AI-Generated Text Detection

Janapati Ganga Sai Ram, Sokalla Aprameya, Mathangi Babji & Dr. P Ashok Babu
Institute of Aeronautical Engineering, Hyderabad, Telangana, India

ABSTRACT: The rise of AI-generated text presents significant challenges in distinguishing human-written text from AI-generated text. We use machine learning algorithm to address this challenge, utilizing the "AI vs. Human Text" dataset, which includes both AI-generated and human-written text. By employing a combination of BERT-based feature extraction and an XGBoost classifier, the system achieves high accuracy in differentiating between the two types of text. The model ensures consistent performance across various contexts. The trained model demonstrates exceptional performance, achieving a 97.7% accuracy rate in classification. This work contributes to the fields of AI ethics and content security by providing a reliable tool to detect AI-generated text, reducing risks across academia, journalism, and social media.
Keywords: Text classification, Machine learning, BERT, Extreme Gradient Boosting (XGBoost), AI text detection

1 INTRODUCTION

Recent advancements in large language models (LLMs) have led to remarkable improvements in natural language understanding (NLU) and natural language generation (NLG). Models such as GPT-3, GPT-3.5, and Llama have set new benchmarks in various natural language processing (NLP) tasks, comprising text summarization, grammar correction, and automated answering. The sophistication of these models allows them to produce text that is virtually indistinguishable from human-written content, raising significant concerns across multiple sectors. The widespread creation of synthetic text generated by these advanced LLMs poses ethical, moral, social, and economic challenges. Furthermore, the potential misuse of these models by cybercriminals for activities such as academic dishonesty, spreading fake news, and conducting phishing attacks emphasises efficient detection methods are needed. Educational institutions, in particular, face a daunting challenge as traditional plagiarism detection tools cannot identify AI-generated content.

As such, novel approaches to differentiate text generated by AI from text written by humans are desperately needed. In order to fill this gap, this work develops a technique for locating distinctive elements and patterns in text produced by artificial intelligence. Our approach uses the power of BERT for feature extraction and the efficiency of extreme Gradient Boosting (XGBoost) for classification. By using a dataset from Kaggle titled "AI Vs Human Text," we develop a model capable of predicting the origin of a given text.

The implementation involves several key steps such as feature extraction using BERT, dimensionality reduction via Principal Component Analysis (PCA), and model training with XGBoost. Our trained model can predict with high accuracy whether a text is AI-generated or human-written. This paper provides a robust detection method, that helps in mitigating the risks associated with the misuse of LLM’s thereby ensuring that these technologies are utilized ethically and effectively.

2 RELATED WORK

Hosam Alamleh et al. proposed a method to classify human-written & ChatGPT-generated text using machine learning. They collected 500 essay and programming responses from students, with ChatGPT generating equivalent responses. Using TF-IDF for feature extraction, they trained eleven models, including Logistic Regression and BERT. Random Forest achieved the highest accuracy at 93% for essays and 93.5% for programming. As the dataset size is small it is challenging for BERT to discriminate between text generated by GPT and text written by humans. [1]

Nuzhat Prova proposed a method to detect AI-generated text using NLP and machine learning. A balanced dataset of 3,000 text samples was created and preprocessed through techniques like stopwords removal and normalization. CountVectorizer was used for feature extraction, and models like BERT, XGBoost, and SVM were trained. At 93% accuracy, BERT outperformed XGBoost (84%), SVM (81%), and other algorithms. However, the relatively small dataset may limit the model's generalizability. [2]

Alicia Tsui Ying Chong et al. proposed a method for detecting machine-generated text (MGT) on Twitter using semantic, emoji, sentiment, and linguistic features. They utilized an enhanced TweepFake dataset and applied a Multi-Layer Perceptron (MLP) classifier along with fine-tuned BERT and emoji2vec embeddings. The model achieved an accuracy of 88.3% in differentiate MGT from human-written text. However, reliance on the TweepFake dataset, which has limitations like missing tweets, may affect the model's generalizability. [3]

Sebastian Gehrmann et al. introduced GLTR, a tool for detecting AI-generated text using statistical analysis. GLTR examines the likelihood of word choices, their rank within the prediction distribution, and the entropy of the word distribution to identify patterns typical of machine-generated content. The tool also provides a visual interface to help users differentiate between human and AI-generated text. It achieved an accuracy of 72%. However, GLTR's reliance on biased sampling assumptions and its short-range memory may limit its effectiveness, especially in adversarial scenarios or longer texts. [4]

Yuhong Mo et al. developed a deep learning model for detecting AI- generated text, utilizing LSTM, Transformer, and CNN layers. The model processes text through extensive preprocessing, followed by embedding into dense vectors and extracting sequence features with a bidirectional LSTM. A custom TransformerBlock captures long- distance dependencies, while Conv1D and GlobalMaxPooling1D layers refine local features into a fixed-length vector. For binary classification, a adam optimizer with a sigmoid activation function is used. However, the models complexity increases computational costs and longer training times. [5]

Vivek Verma et al. proposed Ghostbuster, a method for detecting AI-generated text through a three-stage process. It first computes token probabilities using weaker language models, then generates features via vector and scalar operations, and finally classifies the text with a logistic regression model. The model demonstrated strong detection capabilities but faces challenges with cross-model generalization, with a drop of 6.8 F1 score on unfamiliar models like Claude. [6]

Eric Mitchell et al. introduced DetectGPT, a zero-shot method for detecting machine-generated text by analyzing probability curvature. It identifies AI-generated content by comparing the log probability of original and perturbed text, with a significant drop indicating machine-generated content. DetectGPT requires no labeled data, making it versatile across various domains and models and it's accuracy declines when detecting paraphrased or revised text. [7]

Anton Bakhtin et al. examined the use of Energy-Based Models (EBMs) with training in the residual space of previously trained language models to differentiate between text produced by machines and text created by humans. The EBMs assign lower energy values to human text using binary cross-entropy loss and negative sampling techniques. Tested across datasets like Books and Wikitext, the model showed effectiveness but struggled with generalization, particularly on out-of-domain data. Sensitivity to training domains and architectures limited its performance, as it often learned patterns specific to generated text rather than real text. [8]

Zheng Chen & Huming Liu introduced STADEE, a method combining statistical features and deep learning classifiers to detect machine-generated text. Using a Transformer-Encoder, STADEE processes features like probability and entropy, obtaining an F1 score of 87.05% during in-domain testing, outperforming usual method like GLTR by 9.28%. It also demonstrated strong generalizability with F1 scores of 86.35% in out-of-domain and 82.62% in in-the-wild scenarios, It also struggles in out-of-domain and real-world settings compared to LSTM or InceptionTime

with its generalizability being less robust. [9]

Guangsheng Bao et al. proposed Fast-DetectGPT, a method for detecting machine-generated text using conditional probability curvature. It uses a productive sampling technique in place of DetectGPT's computationally costly perturbation stage. When comparing machine-generated text with human-generated text, Fast- DetectGPT contrasts words. The model accomplishes 340 times faster processing and 75% better detection accuracy. This can be problematic in black-box settings where a single model can't cover all languages and domains. [10]

3 METHODOLOGY

By utilizing BERT for feature extraction and XGBoost for classification, the suggested system seeks to identify text produced by artificial intelligence. The methodology is structured into several key steps: data loading, feature extraction, dimensionality reduction, model training and evaluation.

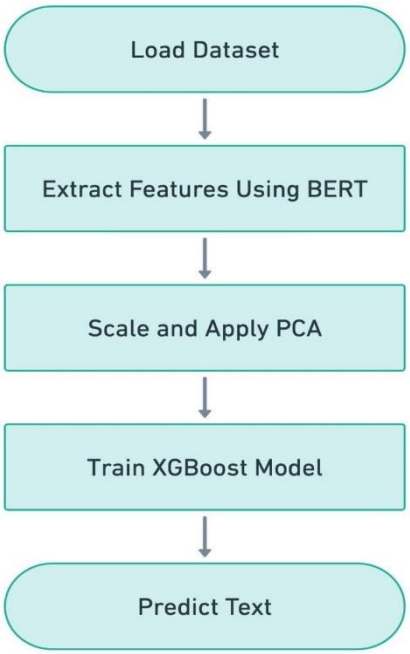

Figure 3.1: Block Diagram of the Proposed Model

3.1 *Loading the Dataset*

The "AI Vs Human Text" dataset from Kaggle, which includes a collection of text samples produced by AI and written by humans, was used in this paper. This dataset includes two columns: Text and Generated, where a value of 0 in the Generated column denotes human-written text, and a value of 1 indicates AI-generated text. Later it uses the pandas library to read the data into a DataFrame and splits it into Text and Generated columns for further processing.

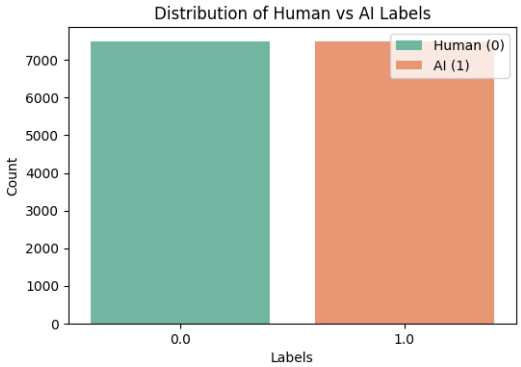

Figure 3.2: Distribution of Human vs AI Labels

3.2 *Feature Extraction Using BERT*

BERT (Bidirectional Encoder Representations from Transformers) is used for feature extraction in order to make use of its potent contextual embeddings. BERT tokenizes the text, capturing both word and subword information, and then processes these tokens through its multiple layers of bidirectional transformers transformer architecture to generate rich semantic embeddings. These embeddings provide a comprehensive representation of the text, capturing nuances beyond simple word frequencies. To produce a fixed-size feature vector for each text sample, mean pooling is applied across all token embeddings, resulting in a comprehensive representation of the text. The resulting machine learning model's capacity to differentiate between text produced by AI and text written by humans is much improved by the high-quality embeddings produced by this technique.

3.3 *Dimensionality Reduction with PCA*

Principal Component Analysis (PCA) is used to handle the high-dimensional feature space produced by BERT to reduce dimensionality, which increases computing efficiency & reduces the risk of overfitting. This involves two main steps: standardization and PCA transformation. To ensure that the PCA method treats all features identically, regardless of their initial scale, the BERT features are first standardised to have a one standard deviation and a zero mean. The majority of variation in the original data is then captured by applying PCA to convert the high-dimensional feature space into a lower-dimensional one, capturing most of the variance present in the original data. In this paper, PCA reduces the dimensionality to 100 components. This reduction helps maintain the essential information needed for classification while improving the model's efficiency. It returns the transformed features, along with the scaler and PCA components used in the process.

3.4 *Model Training with XGBoost*

Model training is performed using XGBoost, a robust gradient boosting algorithm. It takes the PCA-reduced training features and labels as inputs, then it initializes an XGBoost classifier with hyperparameters. The XGBoost model is trained on the training data, ensuring that the trained model is accurately calibrated to differentiate between text authored by humans and text created by AI.

3.5 *Prediction and Model Evaluation*

The trained XGBoost model is used to identify the text as either human-written or AI-generated in real-time by extracting BERT features, and applying the saved scaler and PCA transformations to the input text then the transformed features are given to the XGBoost model for prediction. Several measures are used to assess the model performance on the test data, including F1-score, accuracy, precision & recall. These measures guarantee the system's stability and dependability

in practical applications by offering a thorough picture of how well it detects text generated by artificial intelligence.

4 RESULTS AND DISCUSSION

The proposed approach effectively distinguished either AI-generated text and human-written with high accuracy. The model correctly predicted AI text as AI-generated and human text as human-written, achieving an accuracy of 97.7%. Precision and recall scores were also high, indicating robust performance in both identifying AI text correctly (precision: 98.02%) and capturing a high percentage of actual AI text instances (recall: 97.2%). The F1 score of 97.6% confirms the model's balanced performance by combining precision and recall.

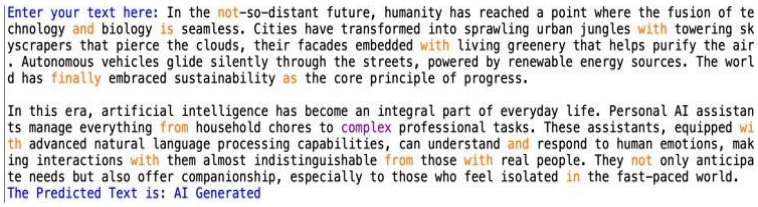

Figure 4.1 Prediction of AI Generated Text

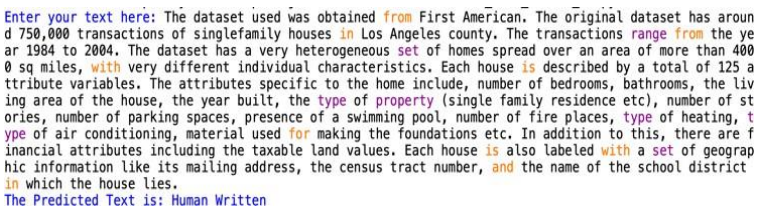

Figure 4.2 Prediction of Human written Text

4.1 *Comparision with other models*

There were significant variations in accuracy and efficacy between the Random Forest classifier and the suggested XGBoost model when it came to differentiating between text created by AI and material authored by humans. With excellent precision (98.0%), recall (97.2%), and F1 score (97.6%), the XGBoost model obtained an impressive accuracy of 97.7%. The robustness of the model in correctly classifying both AI and human text is demonstrated by these metrics.

In comparison, the Random Forest classifier demonstrated somewhat reduced performance metrics. It attained a 96.3% accuracy rate, accompanied with values for precision, recall, and F1 score of 97.1%, 95.3%, and 96.2%, respectively. Although Random Forest performed reasonably well, it showed lower recall compared to XGBoost.

These results underscore the superiority of the XGBoost approach in effectively distinguish human-written text & AI-generated text, outperforming Random Forest in overall accuracy and balance between precision and recall. The combination of BERT-based feature extraction and XGBoost's ensemble learning framework proves advantageous in capturing nuanced differences in text content, thereby enhancing the model's durability & applicability in real-world circumstances requiring precise text classfication.

Table 4.1: Comparision of Model Performance Metrics

Metrics	XGBoost	Random Forest
Accuracy	0.977	0.963
Precision	0.980245	0.971074
Recall	0.972955	0.953347
F1 Score	0.976586	0.962129

4.2 *XGBoost Classifier*

The XGBoost classifier demonstrated strong performance in discerning between text produced by AI and text written by humans. The model's effectiveness in accurately classifying data was demonstrated by the exceptionally high precision, recall, and F1- score for both the AI and human categories.

Table 4.2: Classification report of XGBoost classifier

	Precision	Recall	F1-score	Support
Human	0.973890	0.980934	0.977399	1521
AI	0.980245	0.972955	0.976586	1479
Accuracy	0.977000	0.977000	0.977000	0.977
Macro avg	0.977068	0.976944	0.976993	3000
Weighted avg	0.977023	0.977000	0.976999	3000

4.3 *Random Forest Classifier*

The Random Forest classifier, while still performing well, showed slightly lower metrics compared to XGBoost.

Table 4.3: Classification Report of Random Forest Classifier

	Precision	Recall	F1-score	Support
Human	0.955426	0.972387	0.963832	1521
AI	0.971074	0.953347	0.962129	1479
Accuracy	0.963000	0.963000	0.963000	0.963
Macro avg	0.963250	0.962867	0.962980	3000
Weighted avg	0.963141	0.963000	0.962992	3000

4.4 *Confusion Matrix Analysis*

A detailed analysis of the confusion matrices for each model provides further insights into their performance. The number of true and false values are displayed in the confusion matrix. This matrix provides summary of the performance.

4.4.1 *XGBoost Confusion Matrix*

The XGBoost model demonstrates a good amount of accuracy in differentiating human-generated text & AI text. The model correctly classified 1,492 human texts and 1,439 AI texts. The number of misclassifications (29 false positives and 40 false negatives) is low, indicating that the model is effective at this task.

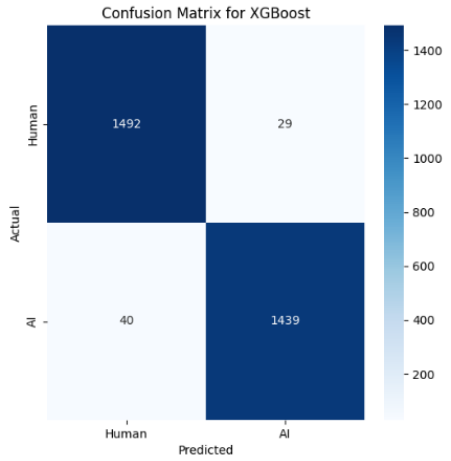

Figure 4.3: Confusion Matrix for XGBoost

4.4.2 *Random Forest Confusion Matrix*

The Random Forest model also performs well but shows slightly higher misclassification rates compared to XGBoost. There are 42 false positives and 69 false negatives. This suggests that while Random Forest is still a strong model for distinguishing AI from human text, it is slightly less accurate than XGBoost.

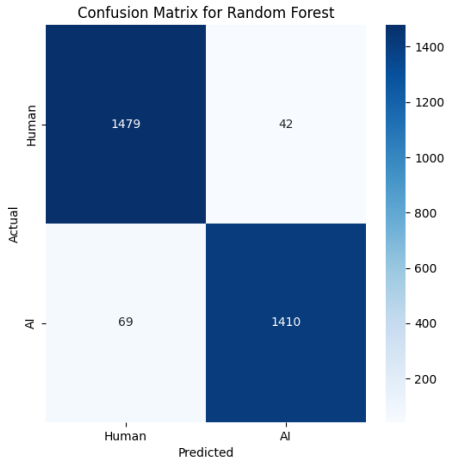

Figure 4.4: Confusion Matrix for Random Forest

Based on the confusion matrix of the two models, XGBoost model is more accurate and precise in classifying human-written text and AI-generated compared to Random Forest model.

5 CONCLUSION

This paper addresses the difficulty of reliably recognizing content generated by artificial intelligence (AI) by developing a strong method to distinguish between text that is written by human or it is generated by AI. Our method leverages BERT for feature extraction and XGBoost for classification, providing a powerful combination for this task. The subtle distinctions between writing by humans and machines are the main problem in identifying text produced by artificial intelligence. By using advanced feature extraction technique and effective dimensionality reduction through PCA, our approach ensures that the model focuses on the most significant features without overwhelming computational complexity.

To enchance the systems's accuracy, we employed comprehensive feature extraction methods.

The result is a highly accurate and efficient text classification system. Our technique attained an outstanding 97.7% accuracy, indicating its usefulness in real-world applications like as content verification and plagiarism detection.

This work offers a workable approach that can be used in a variety of areas, contributing to the expanding field of AI content identification. The results open the door for more study and development in this field by highlighting the potential of integrating sophisticated natural language processing techniques with machine learning classifiers to address the problems presented by AI-generated material.

6 REFERENCES

[1] Alamleh, Hosam, Ali Abdullah S. AlQahtani, and AbdElRahman ElSaid. "Distinguishing human-written and ChatGPT-generated text using machine learning." 2023 Systems and Information Engineering Design Symposium (SIEDS). IEEE, 2023.

[2] Prova, Nuzhat. "Detecting AI Generated Text Based on NLP and Machine Learning Approaches." arXiv preprint arXiv:2404.10032 (2024).

[3] Chong, Alicia Tsui Ying, et al. "Bot or Human? Detection of DeepFake Text with Semantic, Emoji, Sentiment and Linguistic Features." 2023 IEEE 13th International Conference on System Engineering and Technology (ICSET). IEEE, 2023.

[4] Gehrmann, Sebastian, Hendrik Strobelt, and Alexander M. Rush. "Gltr: Statistical detection and visualization of generated text." arXiv preprint arXiv:1906.04043 (2019).

[5] Mo, Yuhong, et al. "Large language model (llm) AI text generation detection based on transformer deep learning algorithm." International Journal of Engineering and Management Research 14.2 (2024): 154-159.).

[6] Verma, Vivek, et al. "Ghostbuster: Detecting text ghostwritten by large language models." arXiv preprint arXiv:2305.15047 (2023).

[7] Mitchell, Eric, et al. "Detectgpt: Zero-shot machine-generated text detection using probability curvature." International Conference on Machine Learning. PMLR, 2023.

[8] Bakhtin, Anton, et al. "Real or fake? learning to discriminate machine from human generated text." arXiv preprint arXiv:1906.03351 (2019).

[9] Chen, Zheng, and Huming Liu. "Stadee: Statistics-based deep detection of machine generated text." International Conference on Intelligent Computing. Singapore: Springer Nature Singapore, 2023.

[10] Bao, Guangsheng, et al. "Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature." arXiv preprint arXiv:2310.05130 (2023).