INTRODUCTION

The rapid advancement of artificial intelligence (AI) in content generation, particularly through models like OpenAI's GPT, has blurred the line between AI-generated and human-authored text. This creates a significant challenge: accurately distinguishing between the two. The task is central to Natural Language Processing (NLP) and has profound implications for ethics, security, and societal trust.

The problem lies in identifying the origin of text. This is crucial for several domains: in **education**, ensuring academic integrity by detecting AI-generated essays; in **media**, verifying the authenticity of news and opinion pieces; and in **cybersecurity**, addressing AI-driven misinformation or phishing campaigns.

The scope of this research focuses on addressing an underexplored area of authorship detection: distinguishing between AI and human text. While traditional authorship studies examine stylistic markers of individual authors, this task involves uncovering nuanced statistical and semantic patterns. Unlike sentiment analysis, where subjective expressions aid classification, this problem demands advanced NLP techniques to tackle its inherent complexity.

The significance of this research is underscored by projections that over 90% of digital content could be AI-generated by 2025 (David Greenberg, 2023). Despite the growth of AI-generation systems, detection tools remain limited in accuracy and scalability. By advancing methods to classify AI-generated text, this study addresses a critical gap, ensuring the trustworthiness and transparency of digital communication in an AI-driven era.

LITERATURE REVIEW

Schaaff et al. (2023) developed a multilingual text classification system to distinguish human- and AI-generated text in English, French, German, and Spanish. They constructed a dataset with 100 human-generated, 100 AI-generated, and 100 AI-rephrased articles per language, covering 10 topics such as politics, sports, and visual arts. The classification approach used XGBoost, Random Forest, and Multilayer Perceptron (MLP) models. Key preprocessing steps included text tokenization, part-of-speech tagging, stopword removal, and sentence embedding using Sentence-BERT. Evaluation metrics such as accuracy and F1-scores revealed impressive performance, with the highest F1-scores reaching 99% for Spanish, 98% for English, 97% for German, and 95% for French. Although the results are highly accurate, the reliance on language-specific feature extraction may limit scalability to unseen languages or evolving AI models.

Islam Prova (2024) explored AI text detection using BERT, SVM, and XGBoost models. The study used a dataset of 3000 text samples equally split between human-written and AI-generated content. Preprocessing steps included stopword removal, stemming, lemmatization, and removing unwanted characters or links. Feature extraction was performed using TF-IDF and CountVectorizer. BERT outperformed the other models, achieving a 93% accuracy, while XGBoost and SVM achieved 84% and 81%, respectively. Despite its effectiveness, the limited dataset size may affect generalization, and the study lacked multilingual or domain-specific evaluation, reducing its applicability across diverse contexts.

Salim and Hossain (2024) created a specialized dataset for human vs AI text classification in the Applied Statistics domain. The dataset consisted of 4231 question-answer pairs collected from 100 students, with corresponding AI-generated answers from ChatGPT. They evaluated models such as DistilBERT, Bert-base, RoBERTa, and ALBERT, using an 80-20 train-test split. Preprocessing steps included plagiarism checks and dataset formatting into JSON and Excel formats. DistilBERT achieved the best performance with an accuracy of 92%, followed by Bert-base at 89%. However, the study's focus on a single academic domain and reliance on ChatGPT limited its generalizability to broader applications.

Islam et al. (2023) proposed a machine learning framework for distinguishing human and AI-generated text using a dataset of 10,000 samples, with 5,204 texts sourced from Quora and CNN and 4,796 ChatGPT-generated paraphrases. Preprocessing steps included TF-IDF vectorization, stopword retention to preserve key features, and binary encoding. Among 11 models tested, the Extremely Randomized Trees Classifier (ERTC) achieved 77% accuracy. The study emphasized ensemble methods' effectiveness but was limited by its reliance on feature-based models, which lack deep contextual understanding and may struggle with diverse datasets.

Kumar et al. (2024) analyzed 700 human and 600 AI-generated responses, leveraging BERT embeddings for feature extraction and cosine similarity for text comparison. Preprocessing included cleaning, similarity scoring, and K-Means clustering, though clusters overlapped due to shared intent in responses. SVM achieved the best performance, with 87% accuracy and an F1-score of 88%. The study was limited by predefined question sets and challenges in cluster separability, potentially restricting its application to unseen data.

Table 1: Literature Comparison

Authors	Advantages	Disadvantages
(Schaaff et al., 2023)	High multilingual classification accuracy	Reliance on language-specific features
(Prova, 2024)	Superior contextual understanding using BERT	Limited dataset size affects generalization
(Salim & Hossain, 2024)	Domain-specific dataset for Applied Statistics	Narrow focus on a single academic domain
(Islam et al., 2023)	Utilized ensemble methods (ERTC) for structured dataset analysis, achieving 77% accuracy.	Reliance on traditional machine learning methods limits scalability to diverse datasets.
(Kumar et al., 2024)	Employed BERT embeddings for feature extraction, high accuracy with SVM (87%).	Challenges with K-Means clustering due to overlapping text features.

SMART OBJECTIVES

- **Specific**

The goal of this project is to classify human-written vs. AI-generated text by implementing two traditional machine learning algorithms (e.g., Logistic Regression & Naïve Bayes) and two deep learning models (e.g., GRU, LSTM). The project will utilize an open-source dataset, and the focus will be on preprocessing, feature extraction, and selecting appropriate models for classification.

- **Measurable**

Success will be determined by achieving an accuracy of 90% or higher, based on benchmarks from similar studies in the literature (ranging from 90-99%).

- **Achievable**

Previous studies using similar datasets structure with machine learning and deep learning models have achieved the required performance level. With well-established preprocessing techniques and model architectures, this goal is realistic. The available dataset is large and diverse, offering a solid foundation for model training.

- **Relevant**

This project is highly relevant in addressing the increasing prevalence of AI-generated content. The ability to detect AI vs. human-written text is crucial in domains like education (academic integrity), media (detecting fake news), and cybersecurity (mitigating AI-driven threats). This work would significantly contribute to the domain of NLP and the broader AI ethics landscape.

- **Time-bound**

Complete dataset preprocessing, model implementation, training, evaluation, and report writing in 9 days:

- ✓ 1 day: Dataset cleaning & feature engineering
- ✓ 1 day: Build traditional ML models
- ✓ 1 day: Build deep learning models
- ✓ 1 day: Hyperparameter tuning of models
- ✓ 1 day: Evaluation, visualization, and analysis
- ✓ 4 days: Report writing

DATASET

➢ Dataset Source

The dataset is sourced from Kaggle, a popular platform for datasets used in machine learning and data science tasks.

➢ Size, Shape, and Structure

The dataset contains 487,235 unique essays. The dataset is in tabular form, where each row represents an individual essay. There are two columns: one for the text of the essay and another for the label (0 for human-written and 1 for AI-generated). The data is textual in nature.

➢ Class Distribution

The dataset is imbalanced, with 305,797 human-written essays (label 0) and 181,438 AI-generated essays (label 1). This imbalance could potentially lead to bias towards the majority class (human-written essays) during model training.

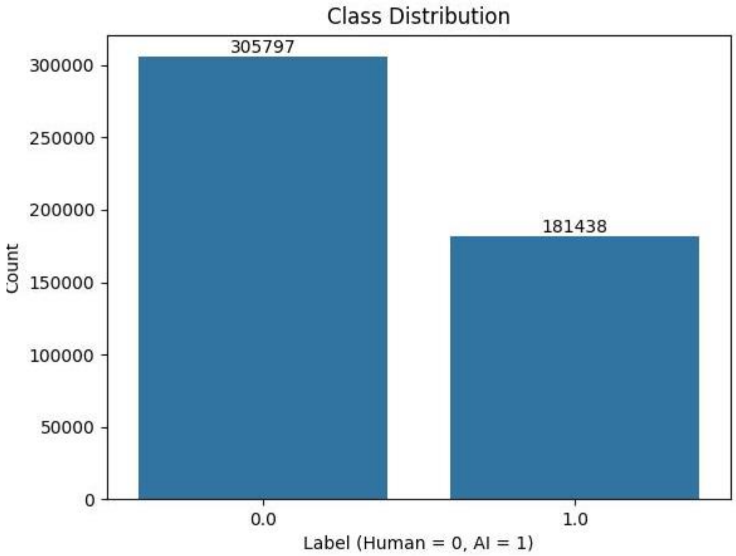

Figure 1: Class Distribution of Dataset

➢ Suitability for the Problem

This dataset is highly suitable for the problem of distinguishing between human-written and AI-generated text, given the large volume of essays and the presence of both classes. The dataset is structured in a way that aligns with typical NLP tasks (text classification) and provides a solid foundation for training machine learning and deep learning models.

Table 2: Strengths & Weaknesses of Dataset

Strengths	Weaknesses
• Large size with diverse essays across various topics ensures robust training. • Well-labeled dataset (ai vs. Human) allows for clear classification tasks. • Textual format is ideal for nlp-based approaches.	• The class imbalance could affect model accuracy and generalization. • The dataset might not cover all potential styles of human writing or AI models, limiting its general applicability.

EXPLORATORY DATA ANALYSIS & PRE-PROCESSING

PREPROCESSING STEPS

1. **Class Imbalance**:

The original dataset of 487,235 entries (305,797 human-written, 181,438 AI-generated) was initially used for analysis. However, due to computational resource limitations, processing the entire dataset resulted in memory errors. To resolve this, 50,000 random entries from each class were selected, balancing the dataset while also reducing the computational load.

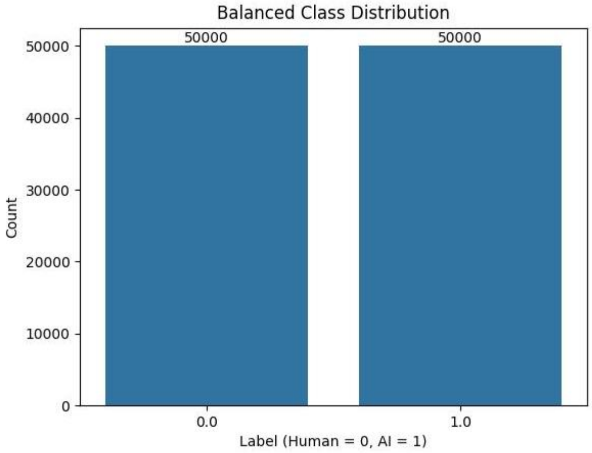

Figure 2: Dataset Classes Post balancing distribution

2. **Text Cleaning**:

- **Stopword Removal**: Commonly occurring but uninformative words (e.g., *the*, *and*, *is*) were removed.
- **Non-Alphabetic Character Removal**: Numbers, punctuation, and other symbols were eliminated.
- **Lowercasing**: Text was converted to lowercase to ensure consistency.

VISUALIZATIONS

1. **Distribution of Essay Length**:
   Histograms of word and character counts highlighted that AI-generated (Label = 1) essays were typically longer and more verbose than human-written (Label = 0) ones.

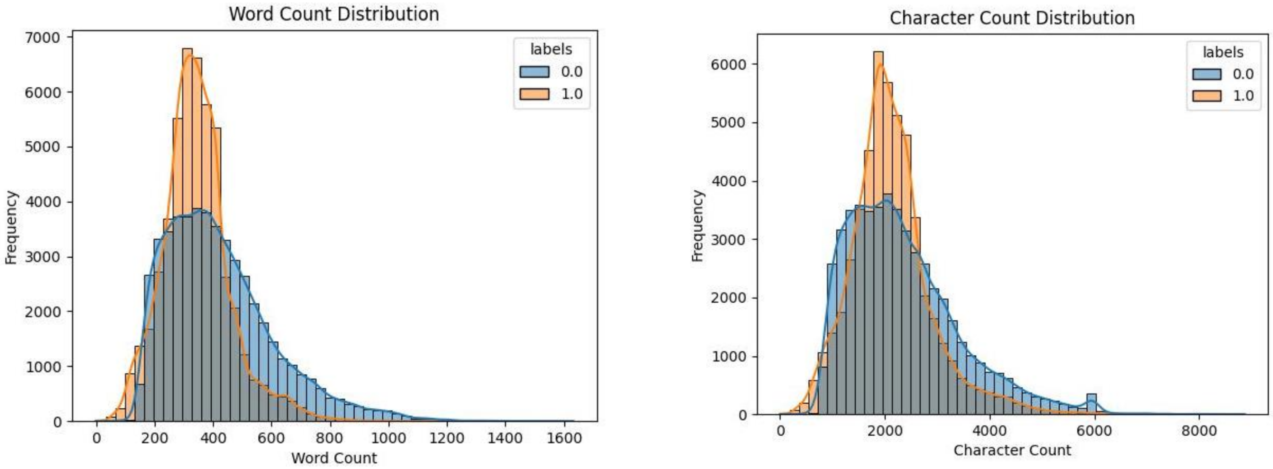

Figure 3: Word & Character Count Distribution of balanced dataset

2. **Word Clouds**:

   - **Human-Written Essays**: Words like *people*, *student*, and *school* dominated the word cloud, suggesting a focus on social and educational topics.

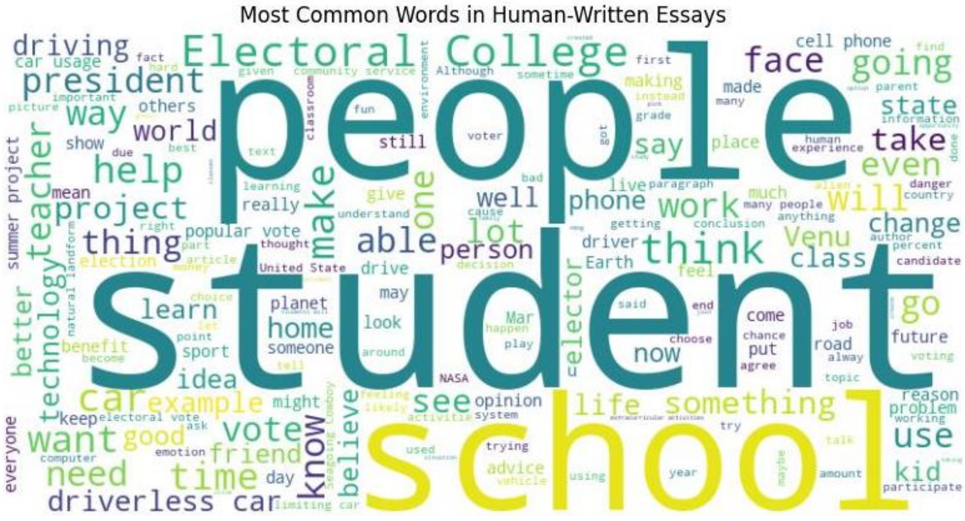

Figure 4: Word Cloud of Human Written Essays

- **AI-Generated Essays**: Word distribution was more balanced, indicating less repetitive patterns compared to human-authored text.

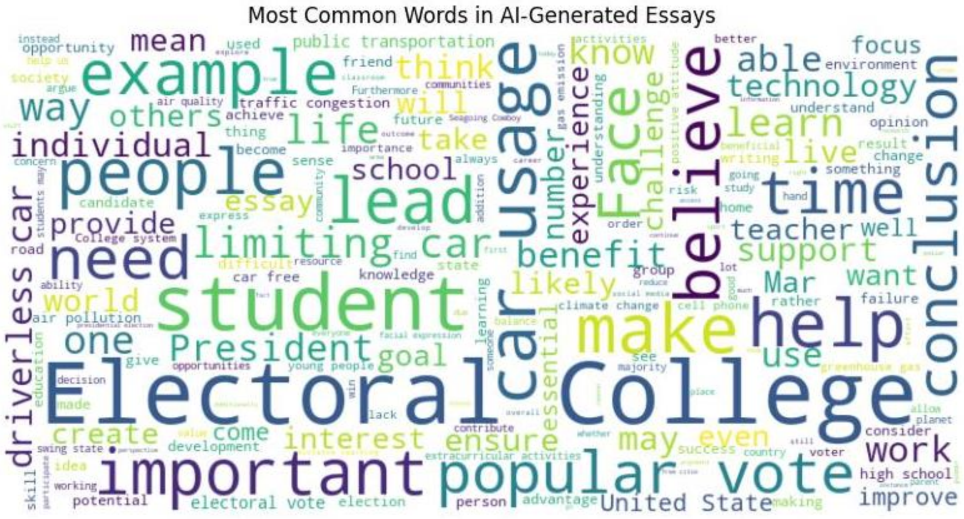
Most Common Words in AI-Generated Essays

Figure 5: Word Cloud of AI Generated Essays

BASELINE

Schaaff et al. (2023) demonstrated classification accuracies up to 99% in multilingual contexts. This study aims for a baseline accuracy of 90%, which is realistic given computational constraints and balanced sampling.

COMPARISON OF TRADITIONAL MACHINE LEARNING MODELS

- **Naive Bayes**
  - ✓ <u>Strengths:</u> Simple and computationally efficient; performs well on text data, particularly when features are independent.
  - ✓ <u>Weaknesses:</u> Assumes feature independence, which may not hold in practice; struggles with complex feature interactions.

- **K-Means (Clustering)**
  - ✓ <u>Strengths:</u> Unsupervised, useful for discovering hidden patterns in data; computationally efficient for large datasets.
  - ✓ <u>Weaknesses:</u> Requires the number of clusters to be predefined; sensitive to outliers; less effective for high-dimensional text data.

- **SVM (Support Vector Machines)**
  - ✓ <u>Strengths:</u> Effective in high-dimensional spaces; robust with various kernels to handle non-linearity.
  - ✓ <u>Weaknesses:</u> Computationally intensive for large datasets; less interpretable compared to simpler models.

- **Logistic Regression**
  - ✓ <u>Strengths:</u> Simple, interpretable, and effective for binary classification tasks; handles large feature sets well with appropriate regularization.
  - ✓ <u>Weaknesses:</u> Linear decision boundary limits its applicability for complex, non-linear problems.

- **Decision Tree**
  - ✓ <u>Strengths:</u> Intuitive and interpretable; handles both categorical and numerical data; no need for feature scaling.
  - ✓ <u>Weaknesses:</u> Prone to overfitting on complex datasets unless pruned; less effective for text data without advanced preprocessing or ensemble methods.

CHOICE OF TRADITIONAL MODELS

- **Naive Bayes**: It is selected due to its simplicity, speed, and strong performance on text data, making it well-suited for the large, text-heavy dataset used in this project. Despite its reliance on the independence assumption, its efficiency makes it a strong candidate for early experimentation.

- **Logistic Regression**: Chosen for its interpretability and ability to serve as a robust baseline model. Its effectiveness in binary classification provides a straightforward comparison to more complex models.

COMPARISON OF DEEP LEARNING MODELS

- **LSTM**
  - ✓ Strengths: Captures long-term dependencies in text; effective for sequential data like language modeling and classification.
  - ✓ Weaknesses: Computationally expensive; struggles with very long sequences due to vanishing gradients.

- **GRU**
  - ✓ Strengths: Simplified architecture compared to LSTM, resulting in faster training; effective for text data with sequential dependencies.
  - ✓ Weaknesses: Can underperform on complex datasets compared to LSTM.

- **BERT**
  - ✓ Strengths: State-of-the-art contextual understanding of text; pretrained on massive corpora; excels in downstream NLP tasks like classification.
  - ✓ Weaknesses: Computationally intensive; requires fine-tuning and significant resources for training.

- **RNN**
  - ✓ Strengths: Effective for sequential data; relatively simpler to implement compared to LSTM/GRU.
  - ✓ Weaknesses: Suffers from vanishing gradient problem, making it less effective for long sequences.

- **CNN**
  - ✓ Strengths: Captures local features effectively; computationally efficient for tasks like sentiment analysis when applied to text.
  - ✓ Weaknesses: Limited in capturing sequential and long-range dependencies in textual data.

CHOICE OF DEEP LEARNING MODELS

- **LSTM**: Selected for its ability to handle long-term dependencies in text, which is crucial for identifying nuanced differences between human and AI-generated essays. Its demonstrated success in text-based tasks supports its suitability for this project.
- **GRU**: GRU offers similar advantages to LSTM but with a simpler architecture, reducing computational overhead. This makes it particularly appealing given the project's dataset size and the need for efficient training.

IMPLEMENTATION

TRADITIONAL MACHINE LEARNING MODELS

➢ LOGISTIC REGRESSION

The text data was first preprocessed using TF-IDF vectorization, limiting the number of features to 5000 to avoid overfitting. The model was trained with a maximum of 100 iterations (max_iter=100). The results were promising, with an accuracy of **98.76%**, far exceeding the benchmark threshold of 90%. To ensure the model was generalizing well and not overfitting, 5-fold cross-validation was performed, resulting in an average accuracy of **98.52%**.

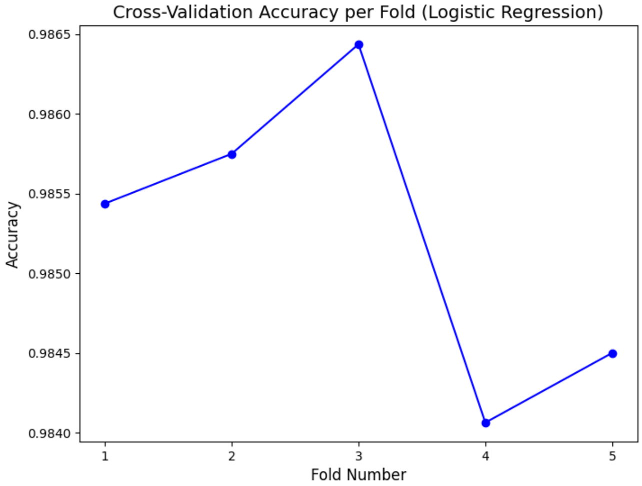

Figure 6: 5-fold cross validation (Logistic Regression)

➢ NAÏVE BAYES

Using TF-IDF transformed data Naïve Bayes was trained. This model performed slightly lower than Logistic Regression, achieving an accuracy of **94.21%**. Cross-validation results were consistent across folds, with an average accuracy of **94.30%.**

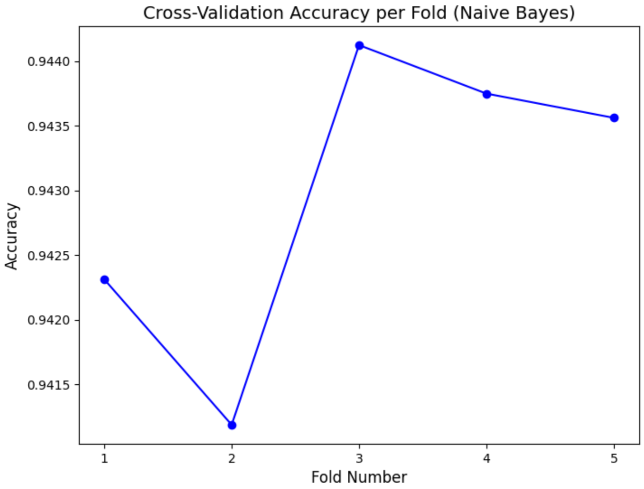

Figure 7: 5-fold cross validation (Naive Bayes)

DEEP LEARNING MODELS

➢ LSTM

the data was tokenized and padded to ensure uniform input lengths. The architecture included an embedding layer with a vocabulary size of 10,000 and an embedding dimension of 128. The LSTM layer had 128 units, followed by a dropout layer with a rate of 0.2 to combat overfitting. A dense layer of 64 units with ReLU activation was used before another dropout layer. The output layer used a sigmoid activation for binary classification. The model was compiled with the Adam optimizer and binary cross-entropy loss, and after training for 5 epochs, it achieved a test accuracy of **98.90%**.

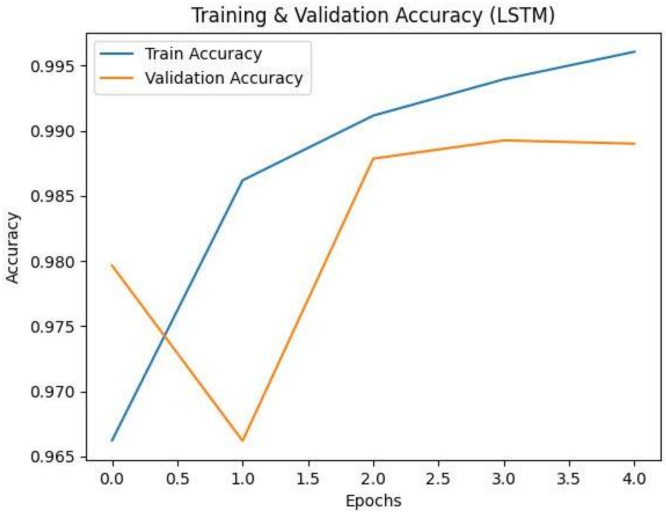

Figure 8: Training & Validation Graph (LSTM)

➢ GRU

For GRU same methodology was implemented as LSTM. The architecture was nearly identical, with the same embedding layer, GRU layer (128 units), and dropout layers. This model performed slightly better than the LSTM, achieving a test accuracy of **98.99%**.

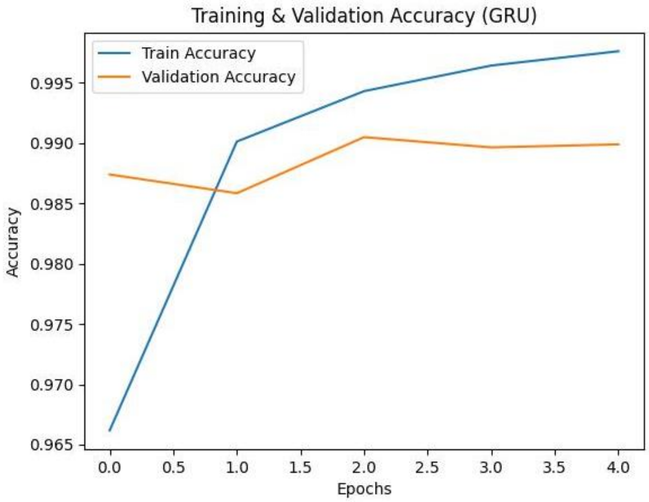

Figure 9: Training & Validation Graph (GRU)

REFINEMENT OF MODELS

➢ LOGISTIC REGRESSION

For Logistic Regression, hyperparameter tuning was a critical step to improve performance. The primary hyperparameters tuned were the **regularization parameter C** and the **maximum number of iterations (max_iter)**. The regularization parameter C controls the strength of regularization; lower values imply stronger regularization, and higher values lead to weaker regularization. We performed a **grid search** across different values for C and max_iter with 5-fold cross-validation. The best results were obtained with C=100 and max_iter=100, leading to an accuracy of **99.25%** after optimization.

➢ NAIVE BAYES

For **Naive Bayes**, hyperparameter tuning focused on the **smoothing parameter alpha**. Smoothing helps adjust the probabilities of unseen words in the training data, ensuring that no probability is zero. **Grid search** was performed on different alpha values and found that **alpha=0.01** provided the best performance. This fine-tuning improved the model's accuracy to **94.35%**.

➢ LSTM AND GRU

For the **LSTM and GRU** models, fine-tuning focused on several aspects due to the computational expense of grid search for deep learning models. We used **manual hyperparameter tuning** to optimize the models. Specifically, we tuned the following:

1. **Number of Neurons**: The number of units in the LSTM/GRU layer was varied, as more neurons may capture more complex relationships, but too many may lead to overfitting.
2. **Dropout Rate**: The dropout rate was adjusted from **0.2 to 0.3**. Dropout helps mitigate overfitting by randomly setting a fraction of input units to 0 during training, improving generalization.
3. **Learning Rate**: The learning rate was varied from **0.001 to 0.0005**. A lower learning rate ensures finer updates to weights, preventing overshooting but may increase training time.
4. **Batch Size**: We varied batch sizes to balance computational efficiency and model performance.

Below Table show best hyperparameter & its corresponding accuracy for LSTM & GRU.

Table 3: Deep learning Models Hyperparameter Tuning Results

LSTM Refinement	Accuracy	GRU Refinement	Accuracy
Baseline LSTM	98.90	**Baseline GRU**	98.99
LSTM Layer Neuron (128->64)	98.43	**GRU Layer Neuron (128->64)**	99.12
Dropout rate (20%->30%)	99.01	**Dropout rate (20%->30%)**	99.12
Learning rate (0.001->0.0005)	99.04	**Learning rate (0.001->0.0005)**	99.01
Batch Size (32->64)	98.76	**Batch Size (32->64)**	99.13

EVALUATION OF TRADITIONAL & DEEP LEARNING MODELS

TRADITIONAL MACHINE LEARNING

➢ Logistic Regression

The base logistic regression model achieved an accuracy of **98.76%** with an F1-score of **0.99** for both classes. The confusion matrix revealed that out of 20,000 samples in the test set, **19,752** were correctly classified. After hyperparameter tuning, accuracy improved to **99.25%**, increasing correctly classified instances to **19,849**. This improvement highlights the model's ability to better generalize across unseen data after optimization.

Table 4: Result Comparison before & after Tuning (Logistic Regression)

Model	Classes	Accuracy	Precision	Recall	F1-Score
Base – logistic	0	98.76	0.98	0.99	0.99
	1		0.99	0.98	0.99
Tuned - logistic	0	99.25	0.99	0.99	0.99
	1		0.99	0.99	0.99

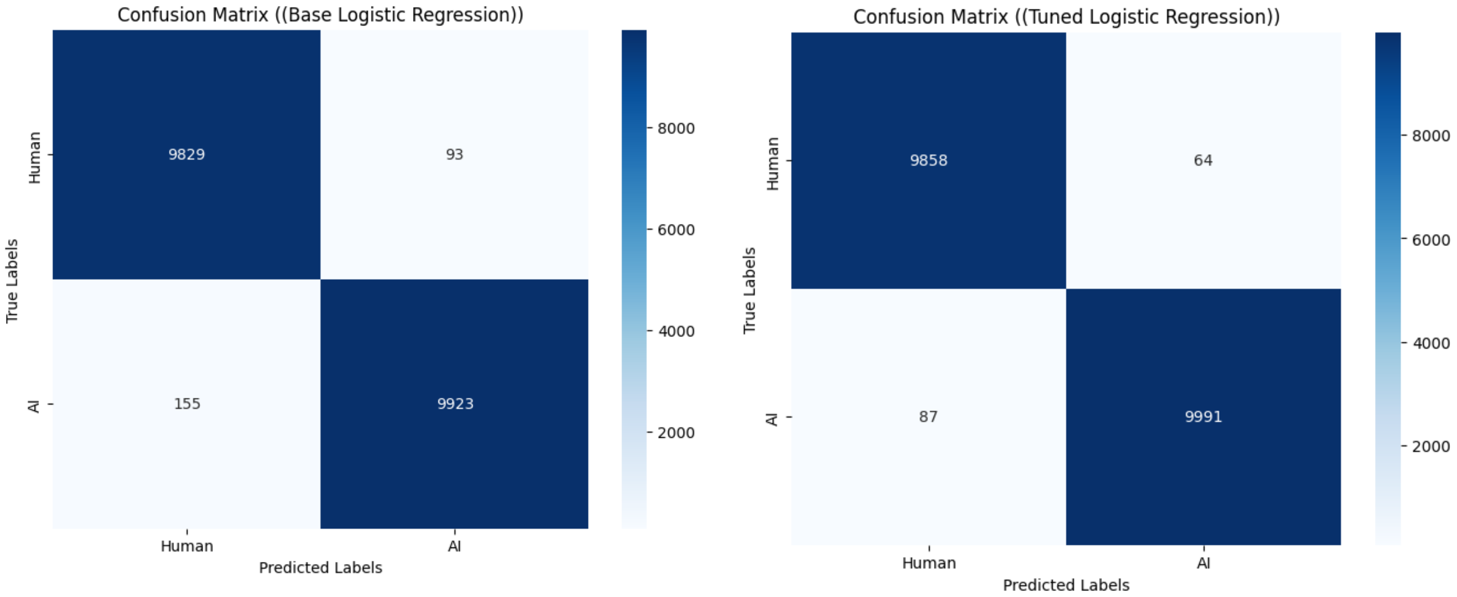

Figure 10: Confusion Matrix before & after hyperparameter tuning (Logistic Regression)

- **Naive Bayes**

The Naive Bayes base model reported an accuracy of **94.21%**, with **18,841** correctly classified instances. Hyperparameter tuning slightly improved the accuracy to **94.35%**, with correctly identified classes increasing to **18,871**. Compared to logistic regression, the improvement in Naive Bayes was marginal, and its performance remained significantly lower, reaffirming logistic regression as the superior traditional model.

Table 5: Result Comparison before & after Tuning (Naive Bayes)

Model	Classes	Accuracy	Precision	Recall	F1-Score
Base – Naïve	0	94.21	0.92	0.96	0.94
	1		0.96	0.92	0.94
Tuned - Naïve	0	94.35	0.93	0.96	0.94
	1		0.96	0.93	0.94

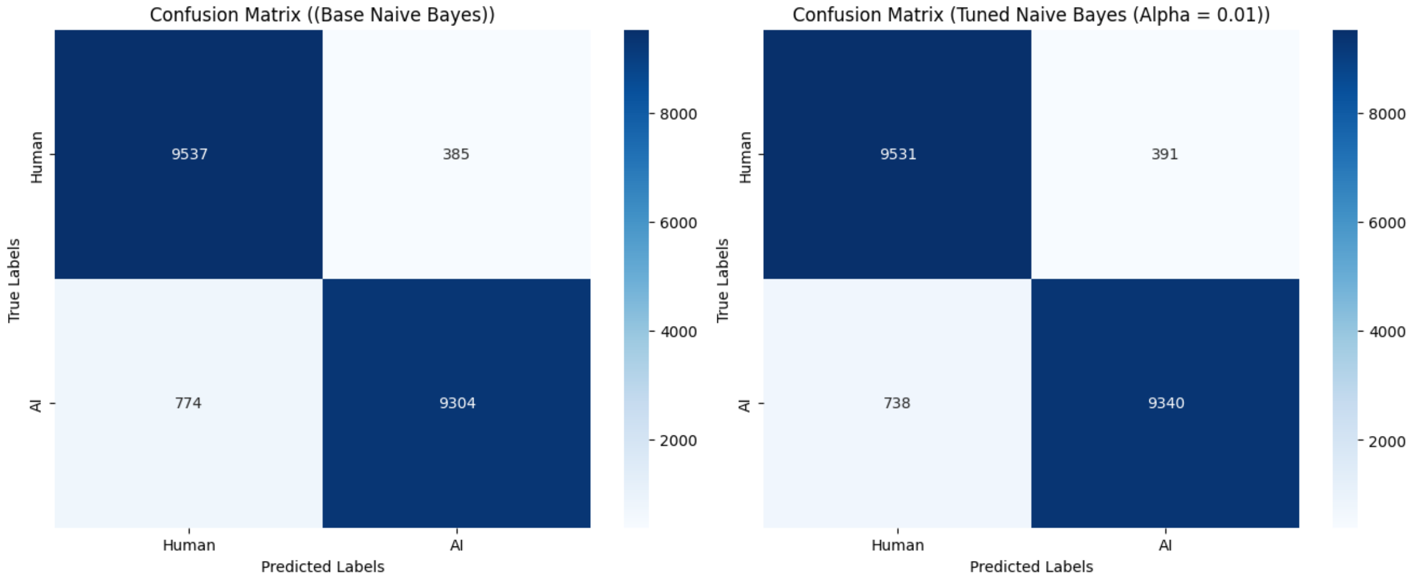

Figure 11: Confusion Matrix before & after hyperparameter tuning (Naive Bayes)

DEEP LEARNING MODELS

➢ LSTM

The base LSTM model demonstrated an accuracy of **98.90%** and an F1-score of **0.99** for both classes. After fine-tuning, which included optimizing dropout rates, learning rate, and the number of neurons, the accuracy increased to **99.04%**, while the F1-score remained stable at **0.99**. Despite the slight increase in accuracy, the fine-tuned model showed consistency and reliability in classifying both human- and AI-generated text.

Table 6: Result Comparison before & after Tuning (LSTM)

Model	Classes	Accuracy	Precision	Recall	F1-Score
Base – LSTM	0	98.90	0.98	0.99	0.99
	1		0.99	0.98	0.99
Tuned - LSTM	0	99.04	0.99	0.99	0.99
	1		0.99	0.99	0.99

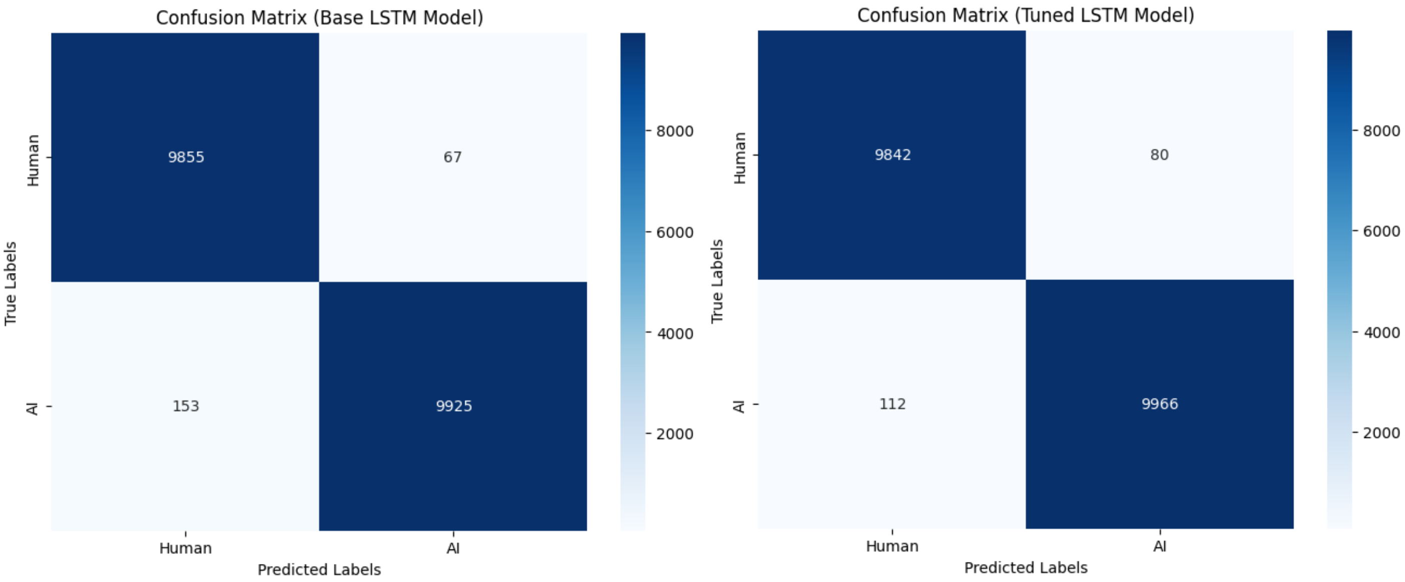

Figure 12: Confusion Matrix before & after hyperparameter tuning (LSTM)

- **GRU**

The GRU model performed exceptionally well, with a base accuracy of **98.99%** and an F1-score of **0.99**. Post-tuning, which involved similar hyperparameter adjustments as in LSTM, the accuracy improved to **99.13%**, with the F1-score unchanged. The refined GRU model slightly outperformed LSTM in terms of accuracy, indicating its better efficiency in capturing dependencies in the text data.

Table 7: Result Comparison before & after Tuning (GRU)

Model	Classes	Accuracy	Precision	Recall	F1-Score
Base – GRU	0	98.99	0.99	0.99	0.99
	1		0.99	0.99	0.99
Tuned - GRU	0	99.13	0.99	0.99	0.99
	1		0.99	0.99	0.99

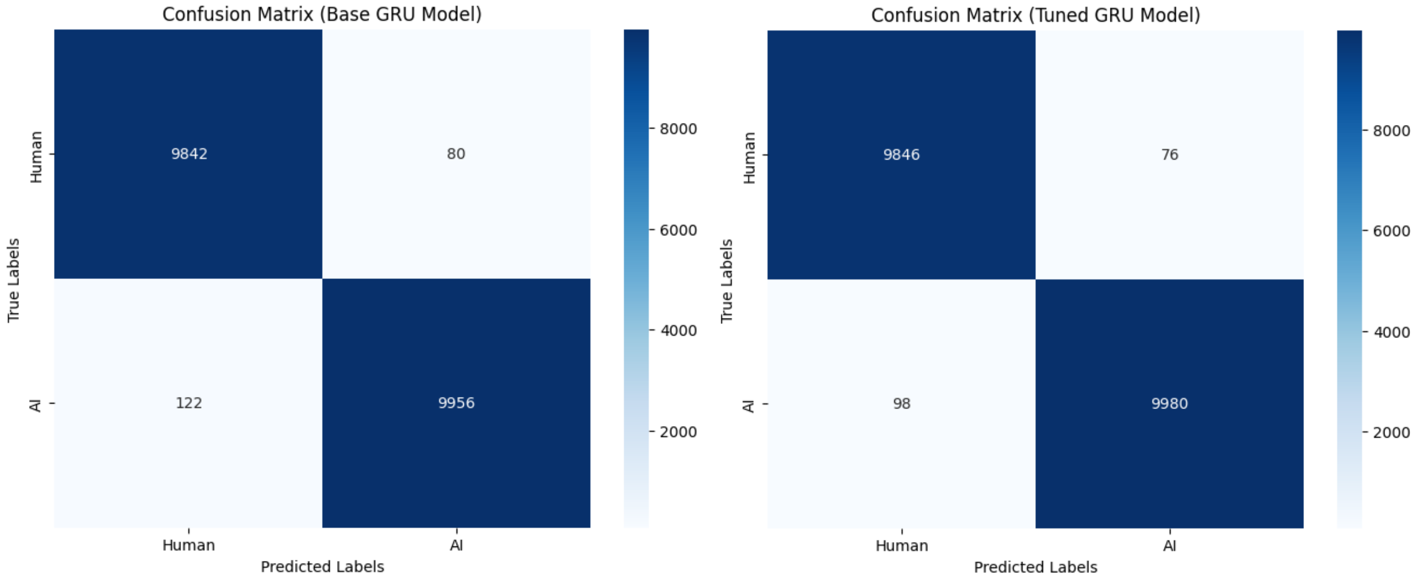

Figure 13: Confusion Matrix before & after hyperparameter tuning (GRU)

CONCLUSION

This study successfully addressed the challenge of distinguishing between human-written and AI-generated text using a balanced dataset and robust classification techniques. Traditional machine learning models like Logistic Regression and Naïve Bayes provided strong performance, with Logistic Regression achieving 99.25% accuracy after optimization. Meanwhile, deep learning models such as LSTM and GRU demonstrated superior capabilities in capturing textual dependencies, with GRU outperforming slightly at 99.13% accuracy. The research highlights the effectiveness of combining advanced preprocessing, appropriate model selection, and hyperparameter tuning to achieve high classification accuracy. These results hold promise for applications in education, media authenticity, and combating misinformation, providing a pathway for scalable and reliable detection systems.

This report uses the **Harvard citation style**, chosen for its clarity, ease of use, and alignment with the study's scholarly approach. Citation styles vary significantly in format, purpose, and disciplinary suitability. **MLA** style emphasizes authorship with its author-page format, favoring literature and humanities. **APA** style, used widely in social sciences, prioritizes the currency of sources through its author-date system. **Chicago** style offers versatility with both notes-bibliography and author-date options, serving a broad academic spectrum. **Harvard** style, also employing the author-date format but with less rigid structuring, is commonly used in the social sciences for its clarity. **Vancouver** style, with numerical citations, excels in technical and medical fields, where concise referencing is critical.

REFERENCES

Schaaff, K., Schlippe, T. and Mindner, L., 2023. Classification of Human-and AI-Generated Texts for English, French, German, and Spanish. *arXiv preprint arXiv:2312.04882*.

Prova, N., 2024. Detecting AI Generated Text Based on NLP and Machine Learning Approaches. *arXiv preprint arXiv:2404.10032*.

Salim, M.S. & Hossain, S.I., 2024. An Applied Statistics dataset for human vs AI-generated answer classification. *Data in Brief*, 54, p.110240. Available at: https://doi.org/10.1016/j.dib.2024.110240 [Accessed 1 December 2024].

Islam, N., Sutradhar, D., Noor, H., Raya, J., Maisha, T. & Farid, D., 2023. Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning. *arXiv*. Available at: https://doi.org/10.48550/arXiv.2306.01761 [Accessed 2 December 2024].

Kumar, S., Tiwari, S., Prasad, R., Rana, A. & M.K., A., 2024. Comparative Analysis of Human and AI Generated Text. *In: Proceedings of the 2024 IEEE SPIN*. pp. 168-173. Available at: https://doi.org/10.1109/SPIN60856.2024.10511301 [Accessed 2 December 2024].

David Greenberg, 2023. *The Daily Financial Trends.* [Online]
Available at: https://www.thefinancialtrends.com/2023/01/16/by-2025-90-of-online-content-could-be-generated-by-ai/?utm_content=cmp-true [Accessed 1 December 2024].