A Comparative Study of Classical Machine Learning Methods for AI-Generated Text Detection Using Statistical Text Features

Imtiaz Hossain
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
imtiaz.hossain@g.bracu.ac.bd
Student ID: 23101137

***Abstract*—The proliferation of large language models (LLMs) such as ChatGPT has raised significant concerns regarding the authenticity and integrity of digital text content. This paper presents a comparative study of classical machine learning methods for detecting AI-generated text using 28 handcrafted statistical text features spanning lexical, syntactic, readability, and distributional categories. I evaluated four classifiers—Support Vector Machine (SVM), AdaBoost, Decision Tree, and Random Forest—on the HC3 (Human ChatGPT Comparison Corpus) dataset and assess cross-dataset generalization on the M4/SemEval-2024 Task 8 dataset. Feature selection is performed using Minimum Redundancy Maximum Relevance (mRMR) and Recursive Feature Elimination (RFE). My results demonstrate that Random Forest achieves the highest in-domain accuracy of 97.40% on HC3, while cross-dataset evaluation reveals substantial domain shift challenges when models trained on ChatGPT-generated text are tested on outputs from different LLM generators. The findings highlight both the discriminative power of statistical text features for single-generator detection and the limitations of feature-based approaches for generalizing across diverse text generators.**

***Index Terms*—AI-generated text detection, machine learning, stylometric analysis, statistical text features, natural language processing**

I. Introduction

The rapid advancement of large language models (LLMs), particularly ChatGPT [1], has enabled the generation of human-like text at unprecedented scale and quality. While these models offer significant productivity benefits, they also pose serious challenges to academic integrity, journalistic credibility, and information trustworthiness [7]. The ability to distinguish between human-written and AI-generated text has therefore become a critical research problem.

Existing detection approaches broadly fall into three categories: (1) zero-shot statistical methods that leverage token probability distributions from language models [2], [15], [17], (2) neural classifier-based approaches that fine-tune pre-trained language models on labeled detection datasets [6], and (3) feature-based methods that extract handcrafted stylometric and statistical features for classification [4], [5]. While neural approaches often achieve high accuracy, they require substantial computational resources and may lack interpretability. In contrast, feature-based methods offer transparency, computational efficiency, and insights into the stylistic differences between human and machine writing.

This paper adopts the feature-based approach, motivated by the hypothesis that AI-generated text exhibits measurable statistical patterns that differ from human writing. Specifically, LLM outputs tend to display more uniform sentence structures, lower lexical diversity, and more predictable word distributions [9], [10]. I designed a comprehensive pipeline that extracts 28 statistical text features, applies principled feature selection, and evaluates four classical machine learning classifiers.

My key contributions are:

- A systematic extraction of 28 statistical text features organized into four interpretable categories: lexical, syntactic, readability, and distributional.
- A comparative evaluation of SVM, AdaBoost, Decision Tree, and Random Forest classifiers with hyperparameter optimization via grid search.
- Cross-dataset generalization analysis using models trained on HC3 (ChatGPT) and evaluated on M4 (multiple LLM generators).
- Feature importance analysis using mRMR and RFE to identify the most discriminative statistical features.

II. Related Work

A. AI-Generated Text Detection

Guo et al. [1] introduced the HC3 dataset, providing paired human and ChatGPT responses across multiple domains, and demonstrated that linguistic features differ measurably between the two sources. Mitchell et al. [2] proposed DetectGPT, a zero-shot method leveraging probability curvature from source models without requiring training data. Gehrmann et al. [3] developed GLTR, which provides statistical visualization tools for detecting generated text based on token likelihood analysis. More recently, Hans et al. [15] introduced Binoculars, a zero-shot detection method using cross-perplexity ratios, while Bao et al. [17] proposed Fast-

DetectGPT with conditional probability curvature for efficient detection.

B. Stylometric and Feature-Based Approaches

Abbasi and Chen [4] pioneered the Writeprints framework for stylometric authorship identification in cyberspace, demonstrating the effectiveness of statistical writing features. Opara [5] extended this approach to AI-generated text detection with StyloAI, using stylometric features to distinguish machine-generated content. Mikhaylovskiy and Churilov [10] analyzed whether LLM-generated text follows Zipf's law similarly to human language, finding statistical divergences. Zaitsu and Jin [11] performed Japanese stylometric analysis to distinguish ChatGPT variants from human writing.

C. Datasets and Benchmarks

Wang et al. [6] organized SemEval-2024 Task 8, establishing a multilingual, multi-generator benchmark for machine-generated text detection. The associated M4 dataset [8] provides texts from multiple generators including ChatGPT, Bloomz, Cohere, and Davinci, enabling cross-generator evaluation. Wu et al. [7] and Valiaiev [20] provide comprehensive surveys of detection methods and future directions.

D. Feature Selection Methods

Peng et al. [14] introduced the mRMR (minimum Redundancy Maximum Relevance) criterion for feature selection, which balances relevance to the target variable against redundancy among selected features. This approach is particularly suitable for high-dimensional feature spaces where correlated features may reduce classifier performance.

III. Methodology

A. System Overview

Fig. 1 illustrates the overall pipeline architecture. The system comprises six sequential stages: data loading and preprocessing, feature extraction, feature selection, classifier training with hyperparameter tuning, evaluation, and visualization.

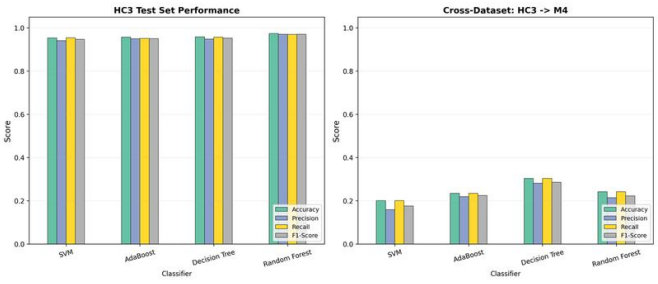

Fig. 1: Classifier performance comparison on HC3 test set (left) and cross-dataset HC3→M4 evaluation (right).

B. Datasets

I utilized two publicly available benchmark datasets:

HC3 (Human ChatGPT Comparison Corpus) [1]: Contains question-answer pairs where each question has both human expert and ChatGPT-generated responses across five domains: open QA, finance, medicine, wiki/CS-AI, and Reddit ELI5. After preprocessing (removing texts shorter than 50 characters), the dataset yields 84,391 texts comprising 57,552 human-written and 26,839 AI-generated samples.

M4/SemEval-2024 Task 8 [6], [8]: A multi-generator dataset containing texts from various LLMs. I used two subsets: (1) the development monolingual set (4,998 texts from Bloomz) for cross-dataset evaluation, and (2) a stratified sample from the training set (10,000 texts across multiple generators including ChatGPT, Bloomz, Cohere, Davinci, and Dolly) for multi-generator analysis.

Table I summarizes the dataset statistics.

TABLE I: Dataset Statistics

Dataset	Total	Human	AI
HC3	84,391	57,552	26,839
M4 Dev (Bloomz)	4,998	2,498	2,500
M4 Train (sampled)	10,000	2,000	8,000

C. Feature Extraction

I extracted 28 statistical text features organized into four categories, as described in Table II. These features capture different aspects of writing style and are designed to be model-agnostic, relying only on surface-level text properties.

Lexical Features (8): Average word length, average sentence length, type-token ratio (vocabulary richness), hapax legomena and dislegomena ratios, Yule's K, Simpson's diversity index, and function word ratio. AI text tends to use narrower vocabulary with more repetitive word choices [10].

Syntactic Features (7): Average punctuation per sentence, comma ratio, question/exclamation/semicolon ratios, conjunction ratio, and pronoun ratio. Punctuation patterns are strong authorship indicators [4].

Readability Features (5): Flesch Reading Ease, Flesch-Kincaid Grade Level, Gunning Fog Index, Coleman-Liau Index, and Automated Readability Index [5].

Distributional Features (8): Stopword, digit, uppercase, and whitespace ratios, average paragraph length, sentence/word length standard deviation, and Zipf's coefficient [9], [10].

Fig. 2 presents the distributions of top-ranked features for human versus AI-generated text. Notable differences are observed in vocabulary richness, where human text shows higher variability, and in sentence length standard deviation, where AI text clusters more tightly around the mean, reflecting the more uniform structure of machine-generated content.

Fig. 3 shows the correlation heatmap across all 28 features. Strong positive correlations exist among readability indices (Flesch-Kincaid, Gunning Fog, ARI), which is expected as

TABLE II: Summary of 28 Statistical Text Features

Category	#	Features
Lexical	8	Avg word/sentence length, TTR, hapax ratios, Yule's K, Simpson's diversity, func. word ratio
Syntactic	7	Punctuation, comma, question, exclamation, semicolon, conj., pronoun
Readability	5	Flesch RE, Flesch-Kincaid, Gunning Fog, Coleman-Liau, ARI
Distributional	8	Stopword, digit, uppercase, whitespace, paragraph len., sent. std, word std, Zipf

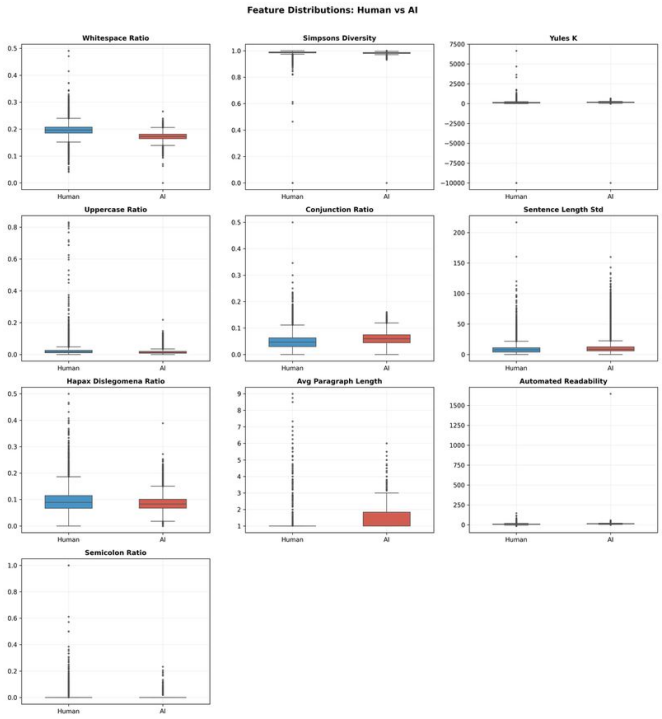

Fig. 2: Box plots comparing the distributions of top features between human-written and AI-generated text on the HC3 dataset.

they measure related complexity aspects. The weaker correlations between lexical and distributional features justify using features from multiple categories to capture complementary information.

D. Feature Selection

Two complementary feature selection methods are employed:

mRMR (Minimum Redundancy Maximum Relevance) [14]: This filter-based method iteratively selects features that maximize mutual information with the target label while minimizing average correlation with previously selected features. The mRMR score for candidate feature f given

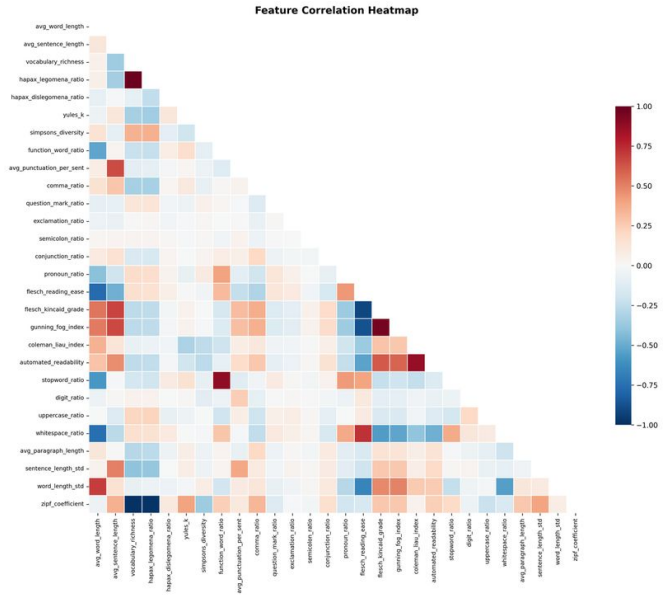

Fig. 3: Correlation heatmap of the 28 statistical text features. Readability features show strong inter-correlation, while cross-category correlations are weaker.

already-selected set S is:

$$\text{mRMR}(f) = I(f;y) - \frac{1}{|S|}\sum_{s\in S}|\rho(f,s)| \tag{1}$$

where $I(f;y)$ is the mutual information between feature f and target y, and $\rho(f,s)$ is the Pearson correlation between features.

RFE (Recursive Feature Elimination): This wrapper-based method uses a linear SVM to iteratively remove the least important feature based on the model's weight coefficients, until the desired number of features remains.

E. Classification

I evaluated four classical machine learning classifiers, each integrated into a standardized pipeline with feature scaling (StandardScaler) and hyperparameter optimization via 5-fold stratified cross-validation GridSearchCV:

Support Vector Machine (SVM): Linear SVM with calibrated probability estimates. Search space: $C \in \{0.1, 1, 10\}$.

AdaBoost: Ensemble boosting with decision stumps. Search space: $n_{\text{estimators}} \in \{50, 100, 200\}$, learning rate $\in \{0.01, 0.1, 1.0\}$.

Decision Tree: CART with configurable depth. Search space: max depth $\in \{5, 10, 20, \text{None}\}$, min samples split $\in \{2, 5, 10\}$, criterion $\in \{\text{gini}, \text{entropy}\}$.

Random Forest: Bagged ensemble of decision trees. Search space: $n_{\text{estimators}} \in \{100, 200\}$, max depth $\in \{10, 20, \text{None}\}$, min samples split $\in \{2, 5\}$.

All models use an 80/20 stratified train-test split on HC3 and are evaluated using accuracy, precision, recall, and macro F1-score.

IV. Results and Discussion

A. Feature Selection Results

The mRMR and RFE methods each selected 15 features. The top mRMR-ranked features by mutual information score include whitespace ratio (0.255), automated readability index (0.227), vocabulary richness (0.227), and Zipf's coefficient (0.212). Fig. 4 shows the feature selection comparison.

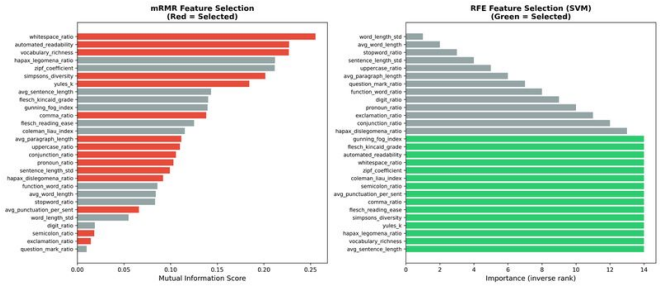

Fig. 4: Feature ranking comparison: mRMR mutual information scores (left, red = selected) and RFE importance via linear SVM (right, green = selected).

The overlap between the two methods includes vocabulary richness, Yule's K, Simpson's diversity, whitespace ratio, automated readability, and comma ratio. The convergence from methodologically different approaches (filter-based mRMR vs. wrapper-based RFE) reinforces the importance of these features.

B. In-Domain Classification (HC3)

Table III presents the classification results on the HC3 test set. Random Forest achieves the best performance across all metrics with 97.40% accuracy and 0.9700 macro F1-score, followed by Decision Tree (95.84%, F1=0.9525), AdaBoost (95.69%, F1=0.9504), and SVM (95.34%, F1=0.9471). All classifiers achieve strong performance, indicating that the 28 statistical features effectively capture distinguishing patterns between human and ChatGPT-generated text.

TABLE III: Classification Results on HC3 Test Set

Classifier	Acc.	Prec.	Rec.	F1
SVM	0.9534	0.9408	0.9544	0.9471
AdaBoost	0.9569	0.9497	0.9511	0.9504
Decision Tree	0.9584	0.9482	0.9572	0.9525
Random Forest	**0.9740**	**0.9702**	**0.9698**	**0.9700**

Table IV shows the optimal hyperparameters identified through grid search for each classifier.

Fig. 5 shows the confusion matrices for all classifiers on the HC3 test set, demonstrating consistently high true positive and true negative rates.

C. Cross-Dataset Generalization

A critical question is whether models trained on HC3 (ChatGPT-only) generalize to text from other LLM generators. Table V presents the cross-dataset evaluation results.

TABLE IV: Optimal Hyperparameters from Grid Search

Classifier	Parameters	CV F1
SVM	$C = 10.0$	0.9463
AdaBoost	$n_{est} = 200$, $lr = 1.0$	0.9505
Decision Tree	depth= 10, split= 5, gini	0.9524
Random Forest	$n_{est} = 200$, split= 2	0.9710

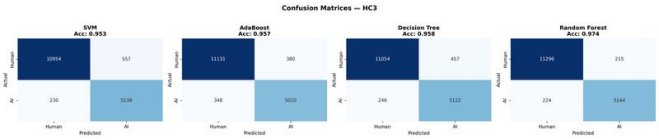

Fig. 5: Confusion matrices for all classifiers on the HC3 test set.

The cross-dataset results reveal a striking generalization gap. On M4 Dev (Bloomz-generated text), all classifiers perform near chance level (20–30% accuracy), indicating that ChatGPT-learned patterns do not transfer to Bloomz. Fig. 7 shows the confusion matrices, where models predominantly classify Bloomz text as human.

On the M4 Train multi-generator subset, performance is higher (76–80% accuracy) due to class imbalance (8,000 AI vs. 2,000 human) and the inclusion of ChatGPT samples. However, F1-scores (0.60–0.66) remain well below in-domain performance. The ROC curves (Fig. 8) further quantify this degradation.

D. Feature Space Visualization

Fig. 9 shows PCA projection onto the first two principal components. The two classes show partial separation along PC1, with overlapping regions explaining the residual 2–5% error.

Fig. 10 presents the t-SNE [16] embedding (perplexity=30). The AI-generated text forms a more compact cluster, consistent with lower stylistic variability, while human text exhibits greater diversity.

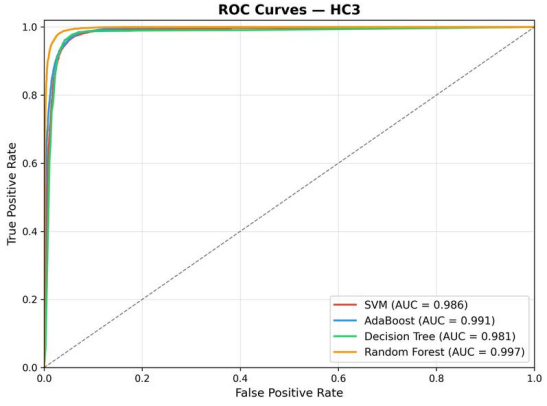

Fig. 6: ROC curves with AUC scores on the HC3 test set.

TABLE V: Cross-Dataset Results: HC3-Trained Models on M4

Dataset	Classifier	Acc.	Prec.	Rec.	F1
M4 Dev (Bloomz)	SVM	0.201	0.160	0.201	0.176
	AdaBoost	0.235	0.220	0.235	0.224
	Dec. Tree	**0.304**	**0.282**	**0.304**	**0.286**
	Rand. Forest	0.242	0.214	0.242	0.223
M4 Train (Multi-gen.)	SVM	**0.800**	**0.673**	0.622	0.637
	AdaBoost	0.766	0.615	0.597	0.603
	Dec. Tree	0.781	0.661	**0.667**	**0.664**
	Rand. Forest	0.761	0.623	0.619	0.621

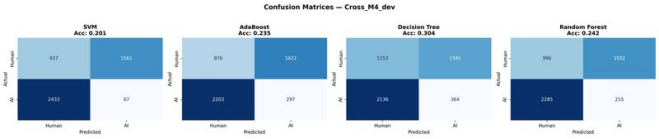

Fig. 7: Confusion matrices for HC3-trained classifiers evaluated on M4 Dev (Bloomz).

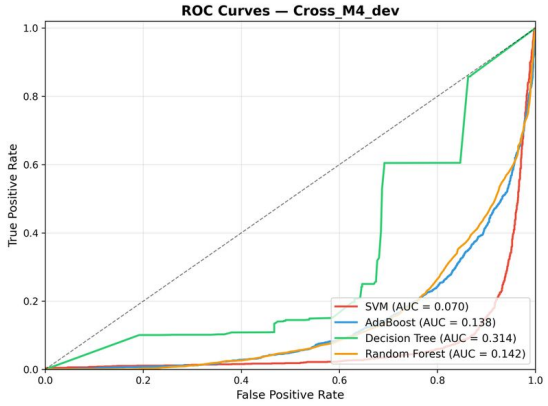

Fig. 8: ROC curves for HC3-trained models on M4 Dev (Bloomz), showing significant AUC degradation.

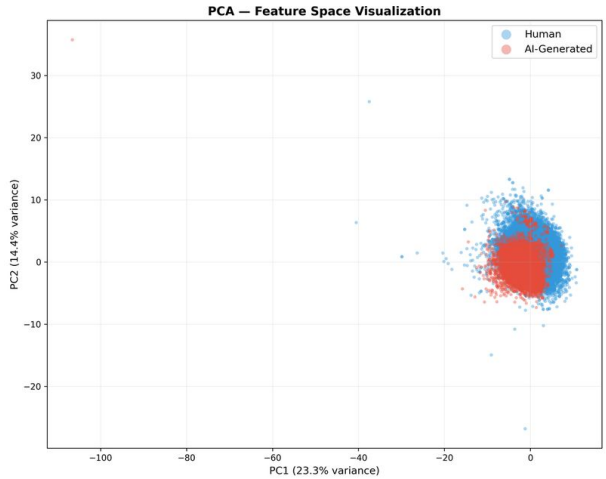

Fig. 9: PCA 2D projection of the feature space. Human (blue) and AI-generated (red) text show partial separation.

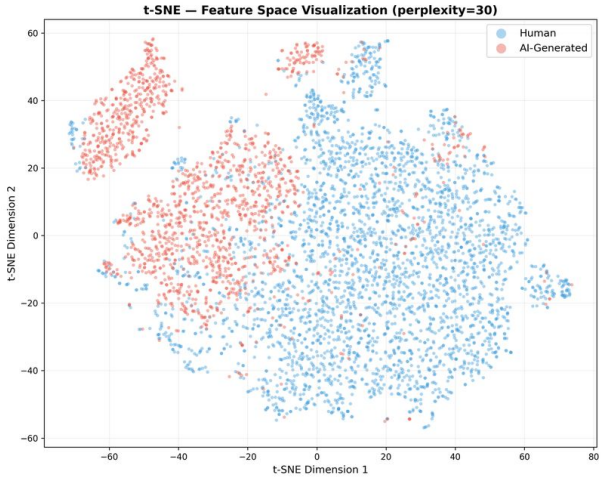

Fig. 10: t-SNE 2D embedding of the feature space. AI text forms tighter clusters than human text.

E. Discussion

The strong in-domain performance (up to 97.40%) demonstrates that statistical text features are effective discriminators when training and test data share the same generator. Random Forest's superiority stems from its ability to model non-linear feature interactions via bootstrap aggregation of 200 decision trees, effectively capturing complex relationships among the 28 features that single classifiers cannot model.

The performance gap between SVM (95.34%) and Random Forest (97.40%) is noteworthy. While SVM with a linear kernel seeks a single optimal hyperplane, Random Forest constructs multiple non-linear decision boundaries, better accommodating the complex feature space geometry visible in the PCA and t-SNE projections (Fig. 9 and 10).

The poor cross-dataset generalization on Bloomz text (20–30%) reveals a fundamental limitation: statistical features capture generator-specific patterns rather than universal human-vs-machine distinctions. Each LLM has different training data, model architectures, and decoding strategies, producing distinct statistical profiles. This aligns with Sadasivan et al. [19] on the theoretical limits of AI text detection, and Tulchinskii et al. [18] who showed that intrinsic dimensionality-based features may offer more robust cross-generator detection.

The feature selection analysis reveals that the top features—whitespace ratio (MI=0.255), vocabulary richness (MI=0.227), automated readability (MI=0.227), and Zipf's coefficient (MI=0.212)—span multiple categories, suggesting AI text has characteristic formatting patterns and word usage distributions. The importance of whitespace ratio indicates that AI text has distinctive paragraph and spacing structures, likely due to the deterministic formatting behavior of LLMs. This is consistent with Mikhaylovskiy and Churilov [10] regarding Zipf's law divergences in LLM-generated text.

Comparing with related work, the 97.40% accuracy achieved by Random Forest on HC3 is competitive with neural approaches reported by Guo et al. [1], while requiring sig-

nificantly fewer computational resources. Unlike transformer-based detectors that need GPU inference, My feature-based pipeline runs efficiently on CPU hardware, making it practical for deployment in resource-constrained environments.

V. Conclusion and Future Work

This paper presented a comprehensive evaluation of classical machine learning methods for AI-generated text detection using 28 statistical text features across four interpretable categories. The key findings are as follows:

- Random Forest achieved the best in-domain accuracy of 97.40% (F1=0.970) on HC3, outperforming SVM, AdaBoost, and Decision Tree.
- Feature selection via mRMR and RFE identified whitespace ratio, vocabulary richness, and Zipf's coefficient as top discriminative features.
- Cross-dataset evaluation revealed a significant generalization gap: 20–30% accuracy on Bloomz-generated text, highlighting generator-specific pattern capture.

Future work should address the generalization gap through: (1) multi-generator training with diverse LLM outputs, (2) domain adaptation techniques to bridge distribution shifts, (3) integration of neural embedding features alongside statistical features for improved robustness, and (4) investigation of temporal drift as LLMs evolve.

Acknowledgment

This work was conducted as part of the CSE437 Machine Learning course at BRAC University. The HC3 dataset was provided by Guo et al. [1] and the M4/SemEval-2024 Task 8 dataset by Wang et al. [6], [8].

References

[1] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, "How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection," *arXiv preprint arXiv:2301.07597*, 2023.

[2] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-shot machine-generated text detection using probability curvature," in *Proc. ICML*, 2023.

[3] S. Gehrmann, H. Strobelt, and A. Rush, "GLTR: Statistical detection and visualization of generated text," in *Proc. ACL*, 2019, pp. 111–116.

[4] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, pp. 1–29, 2008.

[5] I. Opara, "StyloAI: Distinguishing AI-generated content with stylometric analysis," *arXiv preprint arXiv:2401.00920*, 2024.

[6] Y. Wang et al., "SemEval-2024 Task 8: Multidomain, multimodel and multilingual machine-generated text detection," in *Proc. SemEval*, 2024.

[7] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, and L. S. Chao, "A survey on LLM-generated text detection: Necessity, methods, and future directions," *arXiv preprint arXiv:2310.14724*, 2024.

[8] Y. Wang et al., "M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection," in *Proc. EACL*, 2023.

[9] E. Tian, "Technical report on the Pangram AI-generated text classifier," 2023.

[10] N. Mikhaylovskiy and A. Churilov, "Genlangs and Zipf's law: Do languages generated by ChatGPT statistically look human?," *arXiv preprint arXiv:2301.06945*, 2023.

[11] K. Zaitsu and M. Jin, "Distinguishing ChatGPT-generated and human-written papers through Japanese stylometric analysis," *arXiv preprint arXiv:2304.02349*, 2023.

[12] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification: From traditional to deep learning," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–38, 2022.

[13] A. Wahle, N. Madhavji, and M. Steinbacher, "A comparison of SVM against pre-trained language models (PLMs) for text classification tasks," in *Proc. NLPIR*, 2022.

[14] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[15] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein, "Spotting LLMs with Binoculars: Zero-shot detection of machine-generated text," in *Proc. ICML*, 2024.

[16] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[17] G. Bao, Y. Zhao, Z. Teng, L. Yang, and J. Zhang, "Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature," in *Proc. ICLR*, 2024.

[18] E. Tulchinskii et al., "Intrinsic dimension estimation for robust detection of AI-generated texts," in *Proc. NeurIPS*, 2023.

[19] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "On the possibilities of AI-generated text detection," in *Proc. ICML*, 2024.

[20] A. Valiaiev, "Detection of machine-generated text: Literature survey," *arXiv preprint arXiv:2401.02523*, 2024.