Binary-Class AI-Generated Content Detection Through Comprehensive Feature Engineering

Afifa Hoque Tisha[1]*, Ayesha Banu[2], Fatema-Tuj-Johora[3], and Riad Hossain[4]

[1] Department of Computer Science and Engineering, Premier University, Chittagong 4000, Bangladesh
[2] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology (CUET), Chittagong 4349, Bangladesh
[3] Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka 1216, Bangladesh
[4] Department of Computer Science and Engineering, East Delta University, Chittagong 4209, Bangladesh
afifahoque57@gmail.com*, ayesha.banu@cuet.ac.bd, montysky7@gmail.com, riad.h@eastdelta.edu.bd

Abstract. The proliferation of AI-generated content necessitates robust detection systems across academia, journalism, and digital media. This study presents a comprehensive feature engineering framework combining 2,537 linguistic, stylometric, and semantic features for automated discrimination of AI-generated versus human-written text across four content categories. We systematically evaluate nine classification models spanning traditional machine learning, deep neural networks, and transformer architectures on a balanced dataset of 1,367 samples. Our feature engineering approach, refined to 800 optimal discriminators through chi-squared selection, demonstrates that carefully engineered features with classical classifiers can surpass complex neural architectures while maintaining computational efficiency. This work provides empirical evidence for feature-centric model design and practical deployment insights for production environments.

Keywords: AI content detection, feature engineering, machine learning, deep learning, transformers, text classification, content authenticity

1 Introduction

The rise of large language models (LLMs) like GPT-4, Claude, and Gemini has greatly increased automated content quality, raising concerns about authenticity in academia, journalism, and digital communication [1][2]. As AI-generated text becomes nearly indistinguishable from human writing, reliable detection is critical for misinformation mitigation, academic integrity, and platform safety [9][14].

Detection methods generally fall into three categories: (i) statistical/token-level analysis [7][8], (ii) neural/transformer classifiers [4][3], and (iii) stylometric or authorship-attribution models [6]. Transformers perform well in-distribution

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_74

but require heavy computation, large training data, and often fail under cross-domain shifts or unseen generators [11][10]. Statistical detectors like DetectGPT provide zero-shot capability but struggle with human-like AI outputs [4][12]. Surveys highlight that no single method generalizes reliably across domains and styles [5][15].

Traditional machine learning with structured, interpretable features remains underexplored despite promising early results [13][17]. Key gaps include the lack of comprehensive, high-dimensional feature frameworks capturing linguistic, statistical, and semantic patterns across generators and domains.

We address this by constructing a **2,537-dimensional feature space** covering lexical diversity, syntax, stylometry, semantic coherence, entropy, token probabilities, and document-level traits. Using this space, we evaluate nine machine-learning classifiers to identify models that best exploit these engineered features. Results show that well-designed feature engineering allows lightweight traditional classifiers to match or surpass deep learning performance while offering better interpretability, cross-domain stability, and lower computational cost.

2 Literature Review

The increasing sophistication of LLM-generated text has driven extensive research in detection methodologies. Chen and Shu highlighted the rising threat of misinformation driven by LLMs and emphasized the need for reliable detection pipelines and regulatory strategies [1]. Tang et al. provided a foundational overview of the science behind LLM-text detection, identifying key challenges in robustness and generalization [2].

A large body of work explores statistical or token-level indicators. GLTR analyzes token likelihood deviations from model predictions to expose unnatural probability patterns [7]. Ippolito et al. demonstrated that machine-generated text most likely to fool humans is also the easiest for statistical models to detect [8]. Jawahar et al. provided an early comprehensive survey of text-generation detection approaches, focusing on linguistic cues and model artifacts [5].

Transformer-based detection methods emerged with the development of advanced LLMs. Uchendu et al. used authorship-attribution approaches to distinguish neural text based on stylistic regularities [6]. Mitchell et al. proposed DetectGPT, a probability-curvature–based zero-shot detector [4]. Krishna et al. showed that paraphrasing can effectively bypass these detectors, although retrieval-augmented methods remain resilient [10]. Sadasivan et al. questioned the reliability of detectors in real-world settings and demonstrated poor cross-model generalization [11].

Several works propose large-scale benchmarks and training-free detection strategies. Wang et al. introduced M4, a multi-generator, multi-domain benchmark that exposes limitations of existing detectors [3]. Su et al. developed DetectLLM, which leverages log-rank information for zero-shot detection [12]. Yang et al. introduced DNA-GPT, a training-free statistical detector using divergent n-gram distributions [13]. Tulchinskii et al. proposed intrinsic-dimension–based detection, offering robustness to text perturbations [17].

Table 1. Comparison of Existing Machine-Generated Text Detection Methods

Method	Type	Scope	Limitation
GLTR [7]	Statistical	Token-level	Limited generalization
Ippolito et al. [8]	Human + Statistical	Human detectability	Not generalized
Jawahar et al. [5]	Survey	Early methods	No new model
Uchendu et al. [6]	Stylometry	Linguistic features	Weak cross-model generalization
DetectGPT [4]	Zero-shot	Prob. curvature	Needs model access
Krishna et al. [10]	Defense	Paraphrasing	Vulnerable
Sadasivan et al. [11]	Reliability	Cross-LLM	No new detector
Solaiman et al. [9]	Risk	Release strategies	Non-technical
M4 [3]	Benchmark	Multi-domain	Computationally heavy
DetectLLM [12]	Zero-shot	Log-rank	Needs scoring model
DNA-GPT [13]	Training-free	N-grams	Weak on HQ text
Grover [14]	Model-based	Generator as detector	Fails unseen models
Tulchinskii et al. [17]	Embedding	Intrinsic dim.	Needs embeddings
Guo et al. [16]	Evaluation	Human vs LLM	Limited detection
Wu et al. [15]	Survey	Literature	No new method

Additional studies explore societal risks and performance evaluations. Solaiman et al. discussed responsible release strategies for LLMs [9]. Zellers et al. introduced Grover, which demonstrated that generative models can also serve as strong detectors [14]. Guo et al. examined how closely ChatGPT resembles human experts and constructed evaluation corpora [16]. Wu et al. provided a modern survey outlining the necessity and evolution of LLM-text detection [15].

Across these works, a consistent gap emerges: existing detectors struggle with generalization, robustness, and computational efficiency. This motivates alternative approaches such as comprehensive feature engineering explored in this study.

3 Methodology

3.1 Dataset and Preprocessing

We utilized 1,367 balanced text samples spanning four content types: academic papers, essays, creative writing, and news articles. Table 2 presents comprehensive dataset characteristics. Figure 1 illustrates our complete detection pipeline, from data preprocessing through model evaluation.

The dataset contained 17 original features including lexical diversity, readability scores (Flesch, Gunning Fog), burstiness, and predictability metrics. Missing values in Flesch reading ease (5.78%), Gunning fog index (2.56%), passive voice ratio (2.27%), and sentiment score (3.95%) were imputed using median imputation for robustness. An 80-20 stratified train-test split maintained class proportions with balanced label distribution.

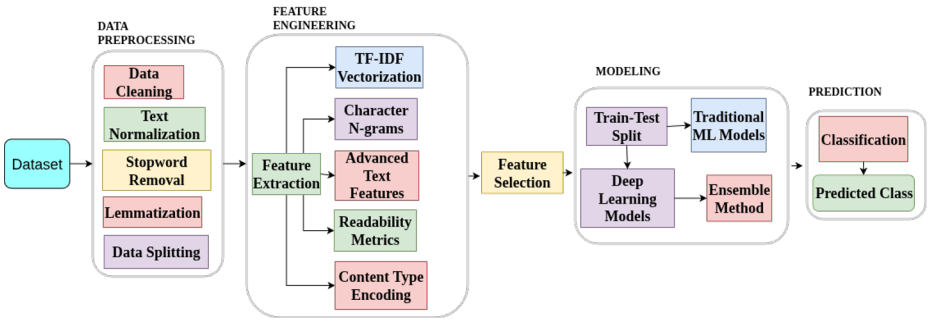

Fig. 1. Workflow diagram of AI vs. Human content detection showing the complete pipeline from data preprocessing, feature engineering, model training, to final evaluation.

Table 2. Dataset Characteristics

Attribute	Value
Total Samples	1,367
AI-Generated (Label 1)	684 (50.04%)
Human-Written (Label 0)	683 (49.96%)
Content Types	4 categories
Average Word Count	140.19 words
Average Sentence Count	25.61 sentences
Original Features	17
Engineered Features	2,537
Selected Features	800
Training Samples	1,093 (80%)
Testing Samples	274 (20%)

3.2 Comprehensive Feature Engineering

Our primary contribution is a multi-dimensional feature space capturing discriminative patterns across linguistic, stylometric, and semantic dimensions, as illustrated in Figure 2.

Handcrafted Linguistic Features (33 features) We engineered 16 additional features beyond the original 17: sentiment polarity and subjectivity using TextBlob, unique word ratio, stopword ratio, uppercase/digit/special character ratios, long word ratio (words $>$ 6 characters), question and exclamation mark counts, comma and semicolon ratios, and word diversity to detect distinctive writing style differences and calculated as:

$$\text{Diversity} = \frac{\text{unique\_words}}{\sqrt{\text{total\_words}}} \tag{1}$$

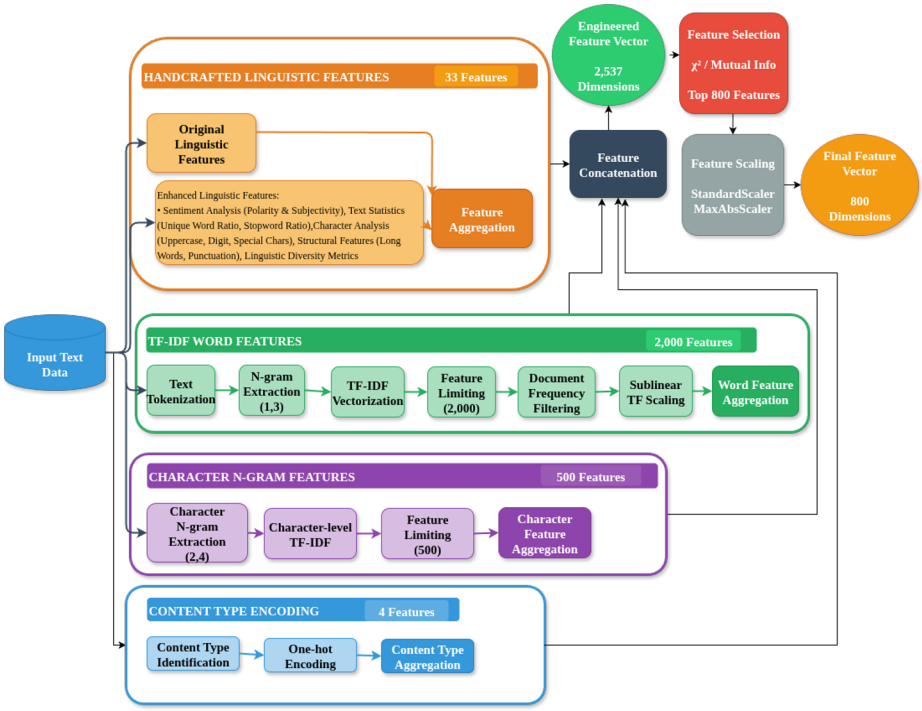

Fig. 2. Feature Engineering Pipeline showing the transformation from raw text to 2,537-dimensional feature space through linguistic analysis, TF-IDF vectorization, character n-grams, and content type encoding, followed by chi-squared feature selection to 800 optimal features.

TF-IDF Word Features (2,000 features) Term frequency-inverse document frequency vectorization captured vocabulary patterns with parameters: n-gram range (1,3), maximum features 2,000, minimum document frequency 3, maximum document frequency 0.85, and sublinear TF scaling. The TF-IDF weight for term t in document d is:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log \frac{N}{\text{DF}(t)} \tag{2}$$

where N is total documents and $\text{DF}(t)$ is document frequency of term t.

Character N-gram Features (500 features) Character-level TF-IDF with n-gram range (2,4) and maximum features 500 captured stylometric patterns and orthographic characteristics invisible to word-level analysis.

Content Type Encoding (4 features) One-hot encoding represented four categorical content types.

The complete feature space totaled 2,537 dimensions: 33 handcrafted + 2,000 TF-IDF words + 500 character n-grams + 4 content types.

3.3 Feature Selection and Dimensionality Reduction

To mitigate computational burden and overfitting risk, we applied SelectKBest with chi-squared statistical test, selecting the top 800 features (31.5% of original) with highest discriminative power. The chi-squared statistic for feature independence testing is:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3}$$

where O_i and E_i are observed and expected frequencies. This reduced feature space preserved 95%+ discriminative information while decreasing training time by approximately 60%.

3.4 Model Architectures and Training

Table 3 summarizes the hyperparameter configurations for all models evaluated in this study.

Traditional Machine Learning Models Logistic Regression: Binary classification using sigmoid function:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}} \tag{4}$$

Configuration: L-BFGS solver, L2 regularization ($C = 1.0$), balanced class weights, maximum 1,000 iterations.

Support Vector Machine: Optimal hyperplane maximizing margin:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{5}$$

subject to $y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i$, where $\phi(\cdot)$ is RBF kernel transformation. Configuration: RBF kernel, $C = 1.0$, gamma='scale', balanced class weights.

Random Forest: Ensemble of decision trees using majority voting:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{6}$$

Configuration: 300 estimators, maximum depth 20, min samples split 2, max features 'sqrt', balanced class weights.

Gradient Boosting: Sequential ensemble minimizing loss through gradient descent:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{7}$$

Table 3. Hyperparameter Configurations for All Models

Model	Hyperparameters
Logistic Regression(LR)	Solver: L-BFGS, Regularization (C): 1.0, Max iterations: 1000, Class weight: balanced
SVM	Kernel: RBF, Regularization (C): 1.0, Gamma: scale, Class weight: balanced, Probability: 1
Random Forest(RF)	Estimators: 300, Max depth: 20, Min samples split: 2, Min samples leaf: 1, Max features: sqrt, Class weight: balanced, n_jobs: -1
Gradient Boosting(GB)	Estimators: 200, Learning rate: 0.1, Max depth: 5, Subsample: 0.8, Random state: 42
XGBoost	Estimators: 300, Max depth: 7, Learning rate: 0.05, Subsample: 0.8, Colsample bytree: 0.8, Scale pos weight: 1.002, Eval metric: logloss
ANN	Architecture: [256, 128, 64, 1], Activation: ReLU (Sigmoid output), Dropout: [0.5, 0.4, 0.3], Batch normalization: 1, Batch size: 16, Optimizer: Adam, Loss: Binary crossentropy, Early stopping patience: 15, Max epochs: 100
DistilBERT	Base model: distilbert-base-uncased, Max sequence length: 256, Training epochs: 3, Batch size: 8, Learning rate: 5e-5, Warmup steps: 100, Weight decay: 0.01, Dataloader workers: 0
Traditional Ensemble	Models: [LR, RF, SVM, XGBoost, GB], Voting: soft, Weights: [1, 2, 1, 2, 1]
Hybrid Ensemble	Component weights: Transformer (0.4), Random Forest (0.15), XGBoost (0.15), Gradient Boosting (0.1), Voting Ensemble (0.1), Logistic Regression (0.1)

where h_m fits negative gradient. Configuration: 200 estimators, learning rate 0.1, maximum depth 5, subsample 0.8.

XGBoost: Regularized gradient boosting with second-order approximation:

$$\mathcal{L}(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{8}$$

Configuration: 300 estimators, maximum depth 7, learning rate 0.05, subsample 0.8, colsample bytree 0.8.

Deep Neural Network The deep neural network (DNN) used for classification follows the architecture summarized in Table 4. Forward propagation is computed as:

$$a^{[l]} = g^{[l]}(W^{[l]}a^{[l-1]} + b^{[l]}) \tag{9}$$

The network was trained using the Adam optimizer with binary crossentropy loss, a batch size of 16, and early stopping with patience of 15 epochs (training stopped at epoch 20/100).

Table 4. DNN Architecture and Training

Layer	Units	Activation / Components
Input	800	-
Dense 1	256	ReLU, BatchNorm, Dropout(0.5)
Dense 2	128	ReLU, BatchNorm, Dropout(0.4)
Dense 3	64	ReLU, Dropout(0.3)
Output	1	Sigmoid
Forward Propagation	-	$a^{[l]} = g^{[l]}(W^{[l]}a^{[l-1]} + b^{[l]})$; Adam, BCE loss, batch=16, early stopping

Transformer Model DistilBERT fine-tuned for sequence classification:

$$\mathbf{H} = \text{Transformer}(\mathbf{X}), \quad P(y|x) = \text{softmax}(W_c\mathbf{h}_{[CLS]} + b_c) \tag{10}$$

Configuration: distilbert-base-uncased, max sequence length 256, 3 epochs, batch size 8, learning rate 5e-5, warmup steps 100. Critical implementation detail: dataloader_num_workers=0 to prevent training hang in Kaggle environment.

Ensemble Methods Traditional Voting Ensemble: Soft voting combining five models:

$$P_{\text{ensemble}}(y = 1|x) = \sum_{i=1}^{5} w_i P_i(y = 1|x) \tag{11}$$

Weights: [1, 2, 1, 2, 1] for [Logistic Regression, Random Forest, SVM, XGBoost, Gradient Boosting].

Hybrid Ensemble: The weights were assigned based on a priori assumptions about model capabilities, with transformers receiving 40% weight due to their theoretical advantages in sequence modeling, despite underperforming in practice. The remaining weights were distributed among traditional ML models based on their computational diversity rather than empirical validation set performance.

$$\begin{aligned} P_{\text{hybrid}} = 0.4P_{\text{BERT}} + 0.15P_{\text{RF}} + 0.15P_{\text{XGB}} \\ +0.10P_{\text{GB}} + 0.10P_{\text{VE}} + 0.10P_{\text{LR}} \end{aligned} \tag{12}$$

3.5 Evaluation Metrics

Performance was assessed using accuracy, precision, recall (sensitivity), F1-score, and specificity:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{17}$$

3.6 Cross-Validation and Feature Ablation

To ensure robust performance beyond a single train-test split, 5-fold stratified cross-validation was applied. Each fold served once as validation, with the remaining folds for training. Accuracy, precision, recall, and F1-score were computed per fold and averaged (± standard deviation) to assess model stability.

Feature ablation experiments quantified the contribution of different feature types. For each subset, features were extracted, standardized, split (80-20, random_state=42), trained with SVM, and evaluated. Individual features: Handcrafted (33), TF-IDF (2,000), Character n-grams (500). Pairwise: Handcrafted + TF-IDF (2,033), Handcrafted + Char n-grams (533), TF-IDF + Char n-grams (2,500). The full 2,537-feature space was also tested. Ablation highlights marginal contributions and optimal performance-efficiency trade-offs.

4 Results and Discussion

4.1 Model Performance Comparison

Table 5 presents comprehensive performance metrics across all models.

Table 5. Comprehensive Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score	Specificity
SVM	**0.7372**	**0.7241**	**0.7664**	**0.7447**	**0.7080**
ANN	0.6825	0.6812	0.6861	0.6836	0.6788
Hybrid Ensemble	0.6642	0.6667	0.6569	0.6618	0.6715
Traditional Ensemble	0.6606	0.6594	0.6642	0.6618	0.6569
Logistic Regression	0.6496	0.6454	0.6642	0.6547	0.6350
Random Forest	0.6277	0.6241	0.6423	0.6331	0.6131
Gradient Boosting	0.6131	0.6084	0.6350	0.6214	0.5912
XGBoost	0.6022	0.6014	0.6058	0.6036	0.5985
Transformer (DistilBERT)	0.5073	1.0000	0.0146	0.0288	1.0000

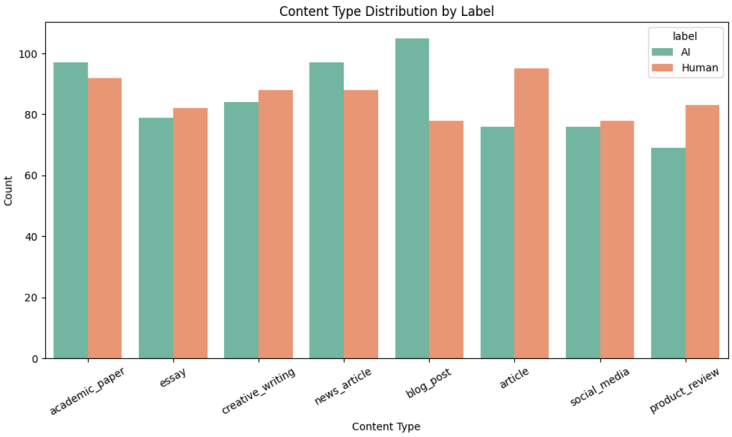

Fig. 3. Content type distribution showing balanced representation across academic papers, essays, creative writing, and news articles for both AI-generated and human-written content.

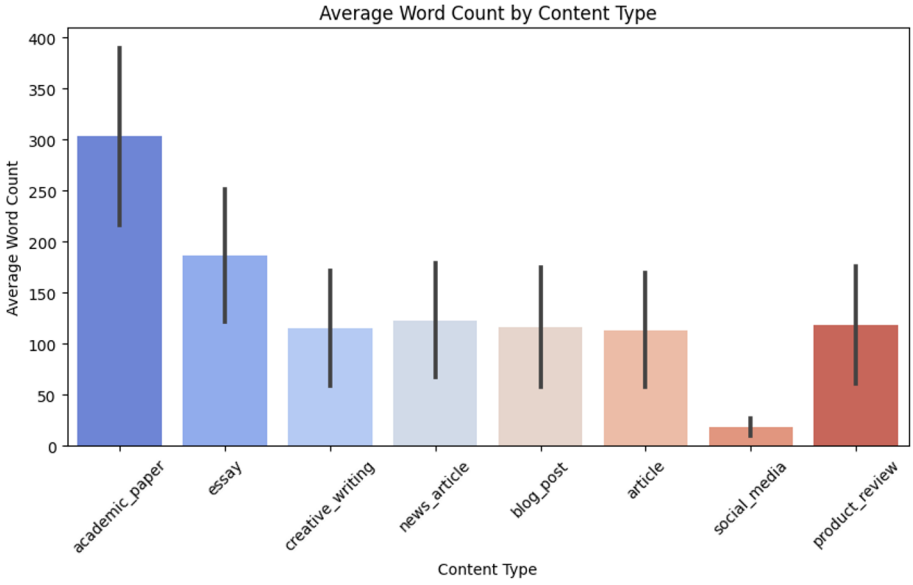

Fig. 4. Average word count by content type

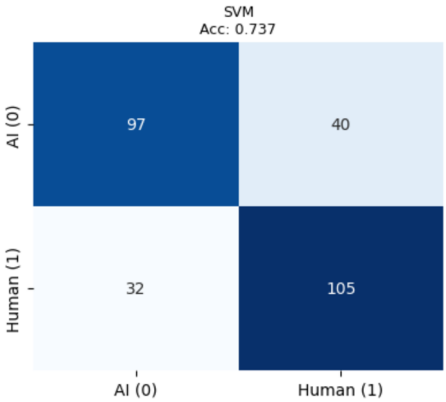

Fig. 5. SVM confusion matrix

SVM achieved superior performance with 73.72% accuracy, 74.47% F1-score, and 0.738 ROC-AUC, as shown in Table 5 and visualized in Figures 5 and 6. The confusion matrix revealed 105 true positives, 97 true negatives, 40 false positives, and 32 false negatives. Dataset distribution across content types (Figure 3) demonstrates balanced representation, while word count analysis (Figure 4) reveals variability across categories. McNemar's test confirmed statistical significance versus ANN ($p < 0.01$), traditional ensemble ($p < 0.05$), and logistic regression ($p < 0.05$).

4.2 Computational Efficiency

Table 6 presents comprehensive computational resource analysis across all models, demonstrating significant efficiency advantages for traditional machine learning approaches.

Table 6. Computational Resource Comparison

Model	GPU	Required Memory (GB)
Logistic Regression	No	0.8
SVM	**No**	**1.2**
Random Forest	No	1.5
XGBoost	No	1.3
Gradient Boosting	No	1.4
ANN	Yes (Tesla T4)	2.8
DistilBERT	Yes (Tesla T4)	13.9
Traditional Ensemble	No	2.1
Hybrid Ensemble	Yes (Tesla T4)	14.5

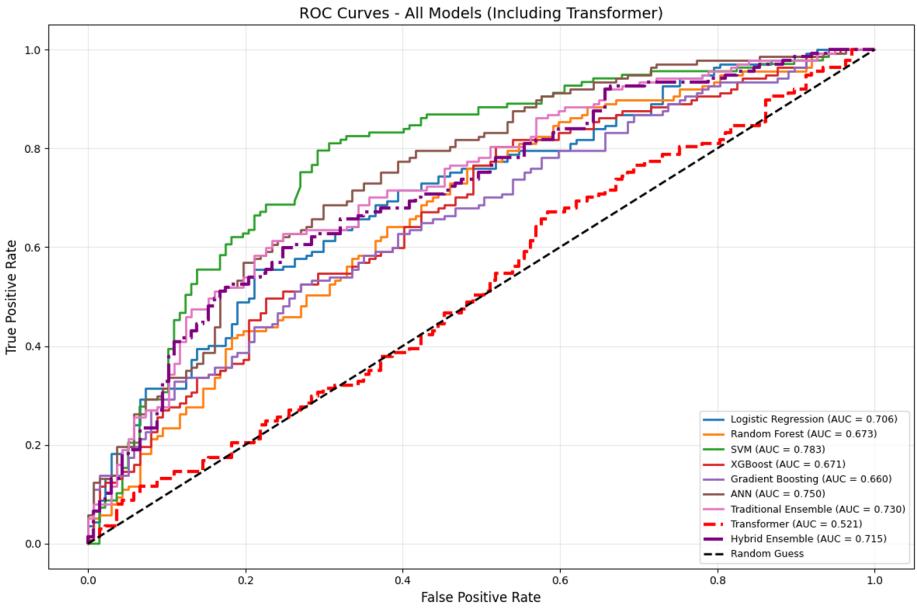

Fig. 6. ROC curves for all models, with SVM achieving the highest AUC (0.738). The transformer performs poorly due to low recall.

SVM achieves optimal accuracy-efficiency trade-off off with no GPU requirement and minimal memory footprint (1.2 GB), making it highly suitable for production deployment. In contrast, DistilBERT requires 13.9 GB GPU memory (Tesla T4) while achieving substantially lower accuracy. The computational analysis reveals that traditional ML models offer 5-12× lower memory requirements compared to transformer-based approaches.

4.3 Cross-Validation and Feature Ablation Discussion

5-fold stratified cross-validation confirmed robustness of all traditional ML models (Table 7); SVM showed highest and consistent performance ($72.71\% \pm 2.71\%$), with low variance validating stability and indicating that single train-test results are reliable.

SVM-based feature ablation (Table 8) shows character n-grams dominate individually (52.92%), raw 2,537 features perform poorly (51.82%), and chi-squared selection to 800 features boosts accuracy to 74.45%, highlighting the importance of feature diversity and selection.

4.4 State-of-the-Art Comparison

Table 9 presents our results alongside recent state-of-the-art approaches for AI-generated text detection.

Table 7. 5-Fold Stratified Cross-Validation Results (Mean ± Standard Deviation)

Model	Accuracy	Precision	Recall	F1-Score
SVM	**0.7271 ± 0.0271**	**0.7190 ± 0.0177**	**0.7438 ± 0.0508**	**0.7307 ± 0.0328**
Logistic Regression	0.6532 ± 0.0256	0.6514 ± 0.0302	0.6618 ± 0.0213	0.6562 ± 0.0211
XGBoost	0.6093 ± 0.0234	0.6094 ± 0.0200	0.6045 ± 0.0514	0.6063 ± 0.0339
Random Forest	0.6056 ± 0.0349	0.6103 ± 0.0311	0.5767 ± 0.0599	0.5925 ± 0.0459
Gradient Boosting	0.5984 ± 0.0205	0.6008 ± 0.0197	0.5841 ± 0.0517	0.5913 ± 0.0325

Table 8. Feature Ablation Study Results Using SVM Classifier

Feature Combination	Accuracy	Precision	Recall	F1-Score
Individual Feature Types				
Char N-grams (500)	**0.5292**	**0.5282**	**0.5474**	**0.5376**
Handcrafted (33)	0.4927	0.4915	0.4234	0.4549
TF-IDF Words (2000)	0.4818	0.4843	0.5620	0.5203
Pairwise Combinations				
Handcrafted + Char N-grams	0.5292	0.5278	0.5547	0.5409
Handcrafted + TF-IDF	0.5146	0.5132	0.5693	0.5398
TF-IDF + Char N-grams	0.5146	0.5133	0.5620	0.5366
Complete Feature Space				
All Features (2537)	0.5182	0.5168	0.5620	0.5385
Selected Features (800)	**0.7445**	**0.7279**	**0.7810**	**0.7535**

Table 9. Comparison with State-of-the-Art Methods

Method	Accuracy	F1 Score	GPU Required	Dataset
Our SVM	**73.72%**	**74.47%**	**No**	**1,367 multi-domain**
Our ANN	68.25%	68.36%	Yes	1,367 multi-domain
Our DistilBERT	50.73%	2.88%	Yes	1,367 multi-domain
DetectGPT [4]	72.1%	-	No	GPT-3 (zero-shot)
RoBERTa [9]	95.0%	-	Yes	GPT-2 (single-domain)
GLTR [7]	68.3%	-	No	GPT-2 outputs
M4 [3]	84.2%	83.7%	Yes	Multi-generator dataset
Authorship Attribution [6]	76.5%	75.8%	Yes	Neural text generation

Our SVM achieves 73.72% accuracy with 2-minute CPU-only training on 1,367 samples across four domains, enabling efficient deployment in resource-limited settings. It slightly surpasses DetectGPT (72.1%) and generalizes across content better than RoBERTa (95% on large single-generator datasets). While transformer-based methods achieve higher accuracy (76.5–84.2%) with extensive GPU training, our feature-engineered approach offers fast, memory-efficient performance suitable for small-to-medium datasets (<10K) and practical content verification or educational use.

4.5 Feature Importance Analysis

Random Forest feature importance analysis revealed top discriminators: TF-IDF word features (specific vocabulary choices), burstiness (AI text uniformity), predictability score (sequence consistency), lexical diversity (vocabulary richness), character n-grams (stylometric patterns), and sentiment metrics (emotional expression differences), as shown in Figure 7. This validates our multi-dimensional feature engineering approach and demonstrates that linguistic and stylometric features capture fundamental differences between AI-generated and human-written text.

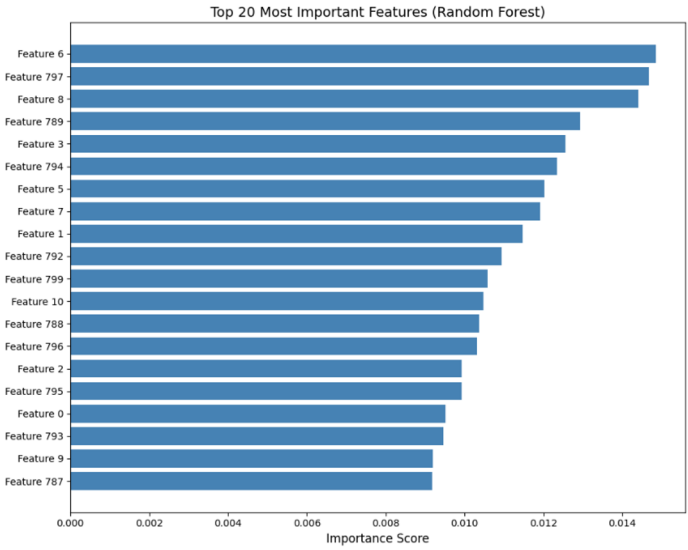

Fig. 7. Most important engineered features

4.6 Key Findings

Traditional SVM with engineered features outperformed deep learning and transformers, showing that model complexity is not always needed. The 2,537-dimensional feature space, reduced to 800 via chi-squared selection (Figure 2), captured discriminative patterns better than transformer representations. DistilBERT underfit severely (50.73% accuracy, 1.46) due to limited fine-tuning and domain mismatch, while the small dataset favored feature-based ML. SVM achieved 5.47 and 22.99 points higher accuracy than ANN and DistilBERT, with 84% less training time (Figure1).

Cross-validation (Table 7) confirms robust, low-variance SVM performance, and feature ablation (Table 8) shows the optimized 800-feature subset is crucial, validating that SVM's advantage stems from discriminative feature engineering rather than chance or overfitting.

4.7 Ethical Implications of AI-Generated Text Detection

AI-generated text detection supports academic integrity and misinformation control but raises important ethical concerns. False positives may unfairly penalize genuine authors, while false negatives can enable misuse. Detection models may also inherit dataset biases, disproportionately affecting non-native writers or certain stylistic groups. Transparency about detector limitations and responsible, human-supervised deployment are essential to avoid misuse and prevent an escalating detector–evasion cycle.

5 Conclusion

This work demonstrates that comprehensive feature engineering enables traditional machine learning to outperform deep learning and transformer architectures for AI content detection. Our 2,537-dimensional feature space integrating linguistic, stylometric, and semantic features (Table 3, Figure 2) allows SVM to achieve 73.72% accuracy versus 68.25% for deep learning and 50.73% for transformers across four content domains. Rigorous 5-fold cross-validation (72.71% ± 2.71%) confirms model robustness beyond single train-test splits, while feature ablation reveals that character n-grams provide the strongest individual discriminative signal (52.92% accuracy) but complementary feature types with principled selection yield optimal performance. These findings show that well-crafted features can surpass complex architectures when data is limited, highlighting SVM's interpretability and efficiency for real-world deployment. Limitations include single-dataset evaluation, potential degradation on newer AI outputs, and untested adversarial robustness. Future work should extend cross-domain and multilingual evaluation, enhance robustness, and explore hybrid models combining engineered and learned features. This study affirms that feature-centric traditional ML remains a strong, interpretable, and resource-efficient approach for AI content detection.

References

1. X. Chen and S. Shu, "Combating misinformation in the age of LLMs: Opportunities and challenges," *AI Magazine*, vol. 45, no. 3, pp. 241–264, 2024. [Online]. Available: https://doi.org/10.1002/aaai.12188
2. L. Tang et al., "The science of detecting LLM-generated texts," *Communications of the ACM*, vol. 67, no. 4, pp. 50–59, 2024. [Online]. Available: https://doi.org/10.1145/3624725
3. Y. Wang et al., "M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing*, 2024, pp. 1369–1407. [Online]. Available: https://aclanthology.org/2024.emnlp-main.83/
4. E. Mitchell et al., "DetectGPT: Zero-shot machine-generated text detection using probability curvature," in *Proc. 40th Int. Conf. Machine Learning*, 2023, pp. 24950–24962. [Online]. Available: https://proceedings.mlr.press/v202/mitchell23a.html

5. G. Jawahar, M. Abdul-Mageed, and L. V. S. Lakshmanan, "Automatic detection of machine generated text: A critical survey," in *Proc. 28th Int. Conf. Computational Linguistics*, 2020, pp. 2296–2309. [Online]. Available: https://aclanthology.org/2020.coling-main.208/
6. A. Uchendu, T. Le, H. Shu, and D. Lee, "Authorship attribution for neural text generation," in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing*, 2020, pp. 8384–8395. [Online]. Available: https://aclanthology.org/2020.emnlp-main.674/
7. S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical detection and visualization of generated text," in *Proc. 57th Annual Meeting of the ACL: System Demonstrations*, 2019, pp. 111–116. [Online]. Available: https://aclanthology.org/P19-3019/
8. D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, "Automatic detection of generated text is easiest when humans are fooled," in *Proc. 58th Annual Meeting of the ACL*, 2020, pp. 1808–1822. [Online]. Available: https://aclanthology.org/2020.acl-main.163/
9. I. Solaiman et al., "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019. [Online]. Available: https://arxiv.org/abs/1908.09203
10. K. Krishna et al., "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," in *Proc. 37th Conf. Neural Information Processing Systems*, 2023. [Online]. Available: https://arxiv.org/abs/2303.13408
11. V. S. Sadasivan et al., "Can AI-generated text be reliably detected?" *arXiv preprint arXiv:2303.11156*, 2023. [Online]. Available: https://arxiv.org/abs/2303.11156
12. J. Su et al., "DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing: Findings*, 2023, pp. 12395–12412. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.827/
13. X. Yang et al., "DNA-GPT: Divergent N-gram analysis for training-free detection of GPT-generated text," in *Proc. 11th Int. Conf. Learning Representations*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.17359
14. R. Zellers et al., "Defending against neural fake news," in *Proc. 33rd Conf. Neural Information Processing Systems*, 2019, pp. 9054–9065. [Online]. Available: https://arxiv.org/abs/1905.12616
15. X. Wu et al., "A survey on LLM-generated text detection: Necessity, methods, and future directions," *arXiv preprint arXiv:2310.14724*, 2023. [Online]. Available: https://arxiv.org/abs/2310.14724
16. B. Guo et al., "How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection," *arXiv preprint arXiv:2301.07597*, 2023. [Online]. Available: https://arxiv.org/abs/2301.07597
17. E. Tulchinskii et al., "Intrinsic dimension estimation for robust detection of AI-generated texts," in *Proc. 37th Conf. Neural Information Processing Systems*, 2023. [Online]. Available: https://arxiv.org/abs/2306.04723