Arabic Text Detection

Abstract

This paper presents a comprehensive framework for distinguishing human-authored Arabic text from AI-generated content using a combination of handcrafted linguistic features and machine-learning models. Several classical classifiers Logistic Regression, Support Vector Machines (SVM), Random Forest, and XGBoost are evaluated alongside a Feedforward Neural Network (FFNN). The proposed feature set captures lexical richness, stylistic regularity, frequency-based patterns, and orthographic cues specific to Arabic. Experimental results on a large-scale Arabic abstract dataset demonstrate that classical ensemble models, particularly Random Forest and XGBoost, significantly outperform neural approaches in this setting, achieving accuracies above 98%. The findings highlight the continued relevance of interpretable, feature-driven models for AI-text detection and provide insights for future Arabic NLP safety applications.

1. Introduction

The emergence of large-scale generative language models has profoundly reshaped the production of Arabic digital text across academic, journalistic, and social media domains. While these models offer substantial benefits, their ability to generate fluent, coherent content raises serious concerns about academic integrity, misinformation, and the misuse of automated content. Consequently, the ability to reliably differentiate human-written text from AI-generated text has become a critical research problem.

Arabic presents unique challenges for AI-text detection due to its rich morphology, flexible word order, orthographic variation, and limited availability of high-quality labeled datasets compared to English. These linguistic characteristics complicate the direct transfer of detection methods developed for other languages.

Recent shared tasks and benchmarks have highlighted the growing importance of Arabic AI-text detection. However, many existing approaches rely heavily on large transformer-based architectures, which can be computationally expensive and difficult to deploy in real-world or resource-constrained environments. This work aims to investigate whether carefully designed linguistic and statistical features, combined with classical machine-learning models, can provide competitive and robust performance.

The primary objective of this study is to systematically evaluate multiple machine-learning models Logistic Regression, SVM, Random Forest, XGBoost, and a Feedforward Neural Network on the task of Arabic AI-text detection, emphasizing interpretability, efficiency, and empirical performance.

2. Related Work

2.1 Arabic AI-Generated Text Dataset

Progress in AI text detection depends heavily on the availability of annotated datasets. Several recent efforts have contributed to Arabic resources for this task. Publicly available datasets derived from academic abstracts and synthetic text generation pipelines have enabled large-scale experimentation. These datasets typically include both original human-authored text and multiple AI-generated variants produced using different language models, allowing for realistic and challenging evaluation scenarios.

Shared evaluation campaigns have further standardized experimental protocols by providing unified benchmarks across diverse text genres. Such efforts have played a critical role in advancing Arabic authorship analysis and detection research.

2.2 Machine Learning Approaches

A substantial body of work has proved that stylometric and lexical features remain effective for AI-generated text detection. Classical models such as Logistic Regression and SVM have been shown to perform competitively when combined with features capturing vocabulary richness, frequency distributions, and structural regularities. These approaches are particularly attractive due to their low computational cost, transparency, and ease of deployment.

In the Arabic context, feature-based classifiers such as Logistic Regression and Support Vector Machines have demonstrated strong performance on short and medium-length texts. Their effectiveness stems from the ability to exploit differences in vocabulary usage, morphological patterns, and orthographic consistency. Studies have shown that AI-generated Arabic text tends to rely more heavily on frequent lexical items, while human-written text exhibits greater lexical richness and structural irregularity (Hamzaouiet al., 2025).

2.3 Ensemble Learning

Ensemble methods such as Random Forest and gradient boosting have been widely adopted in text classification tasks due to their ability to capture non-linear relationships and interactions between features. Prior studies have shown that ensemble models often outperform single classifiers, especially when feature sets include heterogeneous linguistic signals.

Boosting-based methods such as XGBoost further enhance performance by iteratively correcting misclassifications made by previous models. This makes them particularly effective for handling imbalanced datasets and capturing subtle stylistic differences between human and AI-generated text. Prior research has demonstrated that boosted ensembles are highly sensitive to repetitive patterns and frequency artifacts commonly found in machine-generated content (Dong et al., 2020).

2.4 Neural and Transformer Methods

Deep learning models, particularly those based on transformer architectures, have achieved state-of-the-art performance in many NLP tasks. However, recent surveys suggest that neural models may overfit to specific generation patterns and sometimes generalize poorly to unseen generators. Hybrid systems combining neural representations with linguistic features are increasingly viewed as a promising direction.
In AI-generated text detection, transformer-based models excel at capturing long-range dependencies, semantic coherence, and subtle syntactic cues that may not be easily represented by handcrafted features. Fine-tuned transformer classifiers can directly learn discriminative patterns from raw text, reducing the need for extensive feature engineering. Surveys of Arabic NLP research consistently report strong gains when transformer models are applied to classification tasks involving complex linguistic phenomena (Devlin et al., 2019).

3. Dataset Description

The experiments in this study are conducted on the KFUPM-JRCAI Arabic Generated Abstracts dataset. The dataset consists of academic abstracts written by humans and corresponding AI-generated versions produced using multiple language models.

After preprocessing and merging all splits, the final dataset contains 41,940 samples**.** Each instance includes the following fields:

- abstract_text**:** the raw Arabic abstract
- source_split: original dataset partition
- generated_by: source of text generation (human or AI model)
- label**:** binary class label (1 = AI-generated, 0 = human-written)

4. Methodology

4.1 Preprocessing

Arabic text preprocessing was performed to reduce orthographic noise and ensure consistent feature extraction. The preprocessing pipeline included:

- Removal of diacritics (tashkeel)
- Normalization of Hamza forms
- Unification of ligatures (e.g., “ﻵ“ → ”لآ”)
- Removal of excessive punctuation and whitespace
- Tokenization at word and sentence levels

4.2 Feature Engineering

A comprehensive set of handcrafted features was designed to capture lexical diversity, stylistic consistency, and Arabic-specific orthographic patterns. The features include:

- **Hapax Dislegomena Ratio (Feature 14):**
  Measures the proportion of words that occur exactly twice, providing an indicator of rare-word repetition patterns.
- **Average Words per Paragraph (Feature 37):**
  Captures structural regularity and text organization.
- **Top-1000 Frequency Vocabulary Count (Feature 60):**
  Counts words appearing among the most frequent Arabic tokens, reflecting overuse of common vocabulary typical of AI-generated text.
- **Type–Token Ratio (Feature 83):**
  A classical lexical diversity measure indicating vocabulary richness.
- **Tanween Frequency (Feature 106):**
  Measures nunation usage, a subtle orthographic cue often underrepresented in machine-generated Arabic.

4.3 Models

This study evaluates a diverse set of machine-learning and neural models to assess their effectiveness in distinguishing human-written Arabic text from AI-generated content. The selected models represent different learning paradigms, including linear classifiers, margin-based methods, ensemble learning, and neural networks. This diversity enables a comprehensive comparison of interpretability, computational efficiency, and classification performance.

4.3.1 Logistic Regression

Logistic Regression is a probabilistic linear classifier widely used in text classification tasks due to its robustness and interpretability. Given an input feature vector x, Logistic Regression estimates the probability that the instance belongs to the positive class using the logistic (sigmoid) function:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

where $w$represents the learned feature weights and $b$is the bias term.

In the context of AI-generated text detection, Logistic Regression is particularly effective when combined with sparse, high-dimensional representations such as TF-IDF and handcrafted linguistic features. The learned weights allow direct inspection of which features (e.g., lexical diversity or frequency-based metrics) contribute most to the classification decision, making the model suitable for forensic and explainable AI applications.

4.3.2 Support Vector Machine (SVM)

Support Vector Machines are margin-based classifiers that aim to find the optimal hyperplane separating two classes while maximizing the margin between them. For a linear SVM, the optimization objective is defined as:

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

subject to classification constraints, where ξ_i are slack variables and C controls the trade-off between margin maximization and classification error.

In this study, a Linear SVM combined with Truncated Singular Value Decomposition (SVD) is employed to address the high dimensionality of TF-IDF features. Dimensionality reduction projects the original feature space into a lower-dimensional latent semantic space, improving computational efficiency while preserving essential semantic information.

SVMs are known for their strong generalization ability in text classification tasks and have been widely applied to authorship attribution and deception detection. Their robustness makes them well-suited for distinguishing subtle stylistic differences between human and AI-generated Arabic text(Elmitwally,2020).

4.3.3 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees using bootstrap sampling and random feature selection. Each tree independently predicts a class label, and the final decision is obtained via majority voting.

This approach reduces overfitting by decorrelating individual trees and improves robustness when handling noisy or heterogeneous feature sets. Random Forest models are particularly effective when dealing with a mix of lexical, statistical, and structural features, as is the case in Arabic AI-text detection.

From a linguistic perspective, Random Forest can model non-linear interactions between features such as lexical richness, rare-word usage, and orthographic patterns (e.g., Tanween frequency). Prior research has shown that Random Forest performs strongly in stylometric analysis and authorship classification tasks(Elgabry,2021).

4.3.4 XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful boosting-based ensemble method that builds decision trees sequentially, where each new tree corrects the errors of the previous ensemble. The model optimizes a regularized objective function:

$$\mathcal{L} = \sum_{i} l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k)$$

where l is the loss function and Ω penalizes model complexity.

XGBoost is particularly effective in detecting AI-generated text due to its ability to capture fine-grained feature interactions and handle class imbalance. Its regularization mechanisms help prevent overfitting, which is crucial when dealing with large handcrafted feature sets.

Previous NLP studies have reported that gradient boosting models achieve state-of-the-art results in text classification and deception detection tasks, especially when paired with carefully engineered features (Chen & Guestrin, 2016).

4.3.1 Feedforward Neural Network (FFNN)

The Feedforward Neural Network represents a neural approach to the detection task. In this study, the FFNN is trained using dense vector representations of text and optimized using the Adam optimizer with binary cross-entropy loss.

The network consists of:

- An input layer receiving fixed-size embeddings
- One or more hidden dense layers with ReLU activation
- Dropout layers for regularization
- A sigmoid-activated output layer for binary classification

While neural networks excel at learning complex, non-linear decision boundaries, their effectiveness depends heavily on data size, embedding quality, and architectural design. Unlike transformer-based fine-tuning, the FFNN used here relies on fixed representations, which may limit its ability to capture deep contextual nuances.

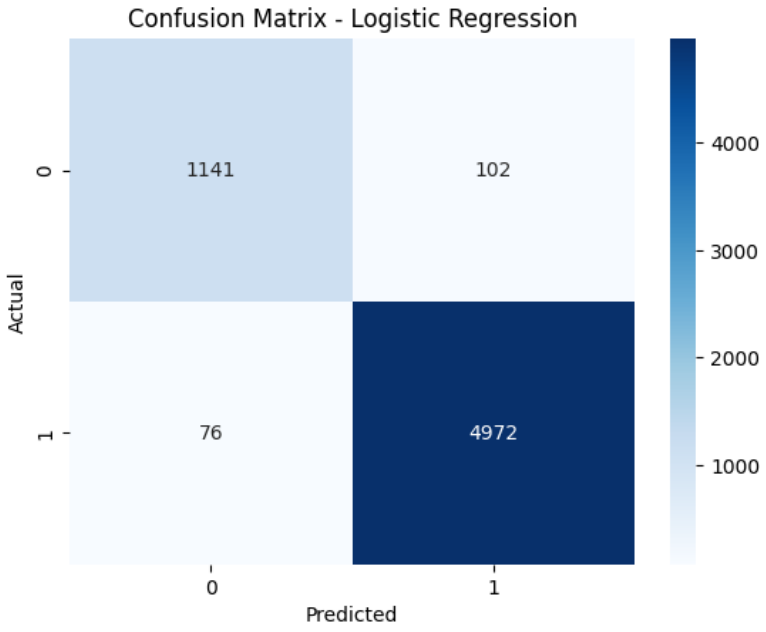
Recent research suggests that shallow neural architectures may underperform compared to ensemble-based classical models for AI-generated text detection, particularly when handcrafted features already encode strong linguistic signals (Uchendu et al., 2021).

5. Results

5.1 Logistic Regression

The Logistic Regression classifier achieved a test accuracy of 97.17%**,** demonstrating strong discriminative power despite its simplicity. The model performed particularly well in identifying AI-generated text, achieving high precision and recall for the majority class.

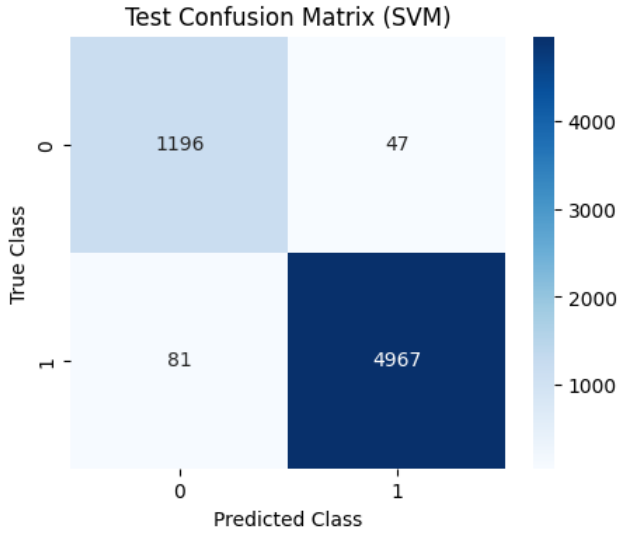
The slightly lower recall for human-written text indicates that some human samples exhibit statistical regularities similar to AI-generated content, especially in formal academic writing. Nevertheless, the overall balance between precision and recall confirms the effectiveness of linear models when paired with rich feature engineering. The Confusion Matrix is as follows.



5.2 Support Vector Machine Results

The SVM classifier improved performance further, achieving an accuracy of 97.97%. The margin-based optimization enabled better separation between the two classes, particularly in ambiguous cases.

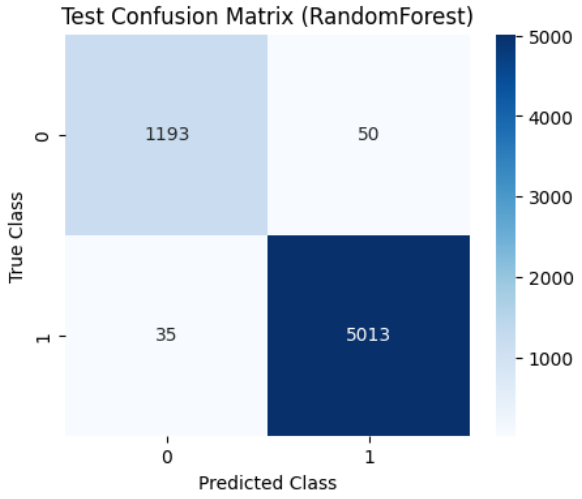
Dimensionality reduction via this model contributed to improved generalization while maintaining semantic structure. This result suggests that latent semantic features remain highly informative for Arabic AI-text detection. The Confusion Matrix is as follows.



5.3 Random Forest

Random Forest achieved the highest overall accuracy of 98.65%, outperforming all other models. The model exhibited strong and balanced precision and recall across both classes.

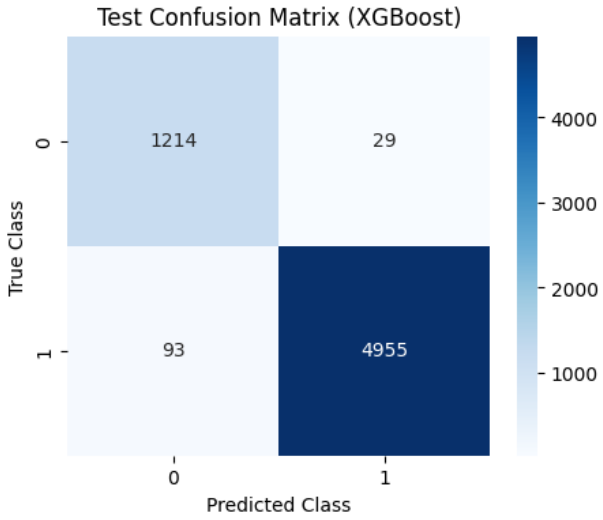
Its superior performance can be attributed to its ability to capture non-linear relationships between linguistic features, such as interactions between lexical diversity measures and frequency-based indicators. This confirms that AI-generated text exhibits complex stylistic patterns that are best modeled using ensemble methods. The Confusion Matrix is as follows.



5.4 XGBoost Results

XGBoost achieved an accuracy **of** 98.06%**,** closely following Random Forest. The model demonstrated particularly high recall for AI-generated text, making it highly sensitive to generation artifacts.

While slightly less accurate than Random Forest, XGBoost offers strong robustness and scalability, making it suitable for large-scale or real-time detection systems.The Confusion Matrix is as follows.



5.5 Feedforward Neural Network

The FFNN achieved a test accuracy of 86.58%**,** significantly lower than classical ensemble models. Although the network performed well in detecting AI-generated samples, it struggled to correctly classify human-written text, resulting in lower recall and F1-score for the minority class.

These findings suggest that shallow neural models may be insufficient for this task without deeper architectures, fine-tuning, or significantly larger training data.

6 Discussion

The summery of models' results in the table below:

Model	Accuracy	Precision (Human)	Recall (Human)	F1-score (Human)	Precision (AI)	Recall (AI)	F1-score (AI)
Logistic Regression	0.9717	0.94	0.92	0.93	0.98	0.98	0.98
(SVM)	0.9797	0.94	0.96	0.95	0.99	0.98	0.99
Random Forest	0.9865	0.97	0.96	0.97	0.99	0.99	0.99
XGBoost	0.9806	0.93	0.98	0.95	0.99	0.98	0.99
(FFNN)	0.8658	0.76	0.47	0.58	0.88	0.96	0.92

The experimental results clearly indicate that feature-driven ensemble models outperform neural approaches in Arabic AI-text detection. Contrary to common assumptions, deep learning does not always guarantee superior performance, particularly when handcrafted linguistic features already capture discriminative stylistic cues.The strong performance of Random Forest and XGBoost underscores the importance of interpretability, robustness, and feature interaction modeling in AI-generated text detection.

7. Conclusion and Future Work

This study investigated the problem of distinguishing human-written Arabic text from AI-generated content using a comprehensive set of linguistic, statistical, and frequency-based features combined with multiple machine-learning and neural models. Given the growing impact of generative language models on Arabic digital content, developing reliable and efficient detection methods has become increasingly important for applications related to academic integrity, content moderation, and AI safety.

Experimental results demonstrate that feature-driven classical machine-learning models outperform neural approaches in this setting. Among all evaluated models, Random Forest achieved the highest accuracy (98.65%)**,** followed closely by XGBoost (98.06%) and Support Vector Machines (97.97%)**.** These models effectively captured non-linear interactions between lexical diversity measures, rare-word usage, frequency-based vocabulary features, and Arabic-specific orthographic cues such as Tanween frequency. The strong performance of these ensemble models highlights their robustness and suitability for Arabic AI-text detection tasks.

Future work can extend this work in several important directions. First, evaluating the proposed detection framework across multiple domains and languages would provide insights into its generalizability beyond academic abstracts and Arabic-only settings. Cross-domain and cross-lingual experiments using multilingual embeddings and transfer learning techniques could enhance robustness against domain shift and unseen text styles. Second, although classical ensemble models achieved strong performance, integrating them with transformer-based representations in hybrid or stacked ensembles may further improve detection accuracy while preserving interpretability. Additionally, exploring automated feature selection and feature discovery methods, such as AutoML or neural architecture search, could reveal latent interactions among linguistic and orthographic features that are not easily captured through manual engineering.

8. References

Hamzaoui, B., Bouchiha, D., & Bouziane, A. (2025). A comprehensive survey on arabic text classification: progress, challenges, and techniques. Brazilian Journal of Technology, 8(1), e77611-e77611.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14(2), 241-258.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).

Elmitwally, N. S. (2020, October). Building a multi-class XGBoost model for Arabic figurative language. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-4). IEEE.

Elgabry, H., Attia, S., Abdel-Rahman, A., Abdel-Ate, A., & Girgis, S. (2021, April). A contextual word embedding for Arabic sarcasm detection with random forests. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 340-344).

Uchendu, A., Le, T., Shu, K., & Lee, D. (2020, November). Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 8384-8395).