

**Detecting AI-Generated Arabic Text Using a Hybrid Approach Integrating Stylistic  
Features and Deep Semantic Embeddings**

Sami Ruzeq M. Almoadawi

Master of Science in Big Data Analytics, Taibah University

MSIS822: Data Analytic Techniques

Dr. Mohammed Al-sarem

December 11, 2025

## Abstract

We are at a critical point where the rapid expansion of Large Language Models (LLMs) poses significant challenges to digital authorship, particularly in Arabic. These models can generate fluent narratives that closely resemble human writing, raising serious concerns about content authenticity and educational integrity. In response, this study develops and evaluates a classification architecture for authenticating Arabic text and distinguishing between human- and AI-generated content. Using the KFUPM-JRCAI dataset, which contains both human and various machine-generated abstracts, a total of 41,940 samples are utilized and split into training, validation, and testing sets (70/15/15) to prevent data leakage (Almutairi et al., 2025). An Arabic-specific preprocessing pipeline is designed, including normalization, removal of non-Arabic characters, stopword filtering, and ISRI stemming. The cleaned text enables the engineering of five stylometric features—multiple elongations, semicolons, interjections, active-voice sentence counts, and BERT [CLS] L2-norm semantic density—to construct a tailored feature space.

To avoid misleading conclusions based on a single aggregate metric, three complementary model-selection schemes are explored: (i) a custom weighted average over global metrics, (ii) Macro F1-Score (average F1 over both classes), and (iii) a comprehensive average over six key measures (Accuracy, ROC-AUC, Macro F1, AI F1, Human F1, and Human Recall). Across all three criteria, the Random Forest classifier consistently emerges as the best overall compromise, achieving strong F1-Scores for both AI-generated and human-written texts. A detailed per-class analysis, however, shows that some high-performing models (notably the BERT-based neural network) can still be

heavily biased toward the AI class, underscoring the importance of per-class evaluation when deploying AI-text detectors in academic settings.

## **Introduction**

The recent advances in Large Language Models (LLMs) have made it remarkably easy to generate fluent Arabic text, including research abstracts and academic assignments. While these models can be useful for drafting and language support, their uncontrolled use in educational contexts raises serious concerns about academic integrity, authorship, and the trustworthiness of submitted work. This tension is particularly acute in Arabic, where institutional guidelines and technical tools for detecting AI-generated content are still emerging (Cheng et al., 2006).

Existing AI-text detectors are predominantly designed and evaluated for English and often fail to account for the morphological, orthographic, and stylistic characteristics of Modern Standard Arabic. Furthermore, many prior studies report performance using a single global metric, such as overall accuracy or F1-score, which can obscure large discrepancies between the AI-generated and human-written classes. For instance, a detector may achieve a high overall F1-score while still misclassifying a substantial portion of genuine human texts as AI-generated, a failure mode that is particularly problematic in academic settings.

This project investigates the capability of supervised AI classifiers to distinguish between human-written and AI-generated Arabic research abstracts. The work is based on the KFUPM-JRCAI Arabic-generated-abstracts dataset, which pairs original human abstracts with machine-generated versions created through by-polishing, from-title, and from-title-and-content generation modes. Building on this dataset, the project designs an

Arabic-specific preprocessing pipeline, derives a focused set of stylometric features together with a BERT-derived semantic signal, and trains several classification models under a fixed 70/15/15 train–validation–test split (Shu & Ye, 2022).

The study has three main objectives: (1) to construct an end-to-end data mining pipeline that operationalizes AI-based detection of Arabic AI-generated text; (2) to quantify how informative a targeted subset of stylometric and BERT-based features is for separating human and machine abstracts; and (3) to compare multiple classifiers under several complementary model-selection schemes, including a custom weighted combination of accuracy, precision, recall, F1-score, ROC-AUC, and Macro F1-Score, and a comprehensive average of key per-class and global metrics, thereby obtaining a principled measure of each model’s overall detection strength.

More broadly, this study aims to contribute to the emerging line of research on AI-text detection in non-English languages, with a particular emphasis on the linguistic and orthographic challenges that arise in Arabic. By analyzing both model-level performance and class-specific behavior, the project also seeks to highlight the limitations and potential biases of current detection approaches, thereby informing future research and practical deployment.

## **Related Work**

Research on AI-generated text detection has accelerated with the advent of large language models (LLMs) such as GPT-3 and GPT-4. Early approaches relied on stylometric features and perplexity-based heuristics or on classifiers using shallow lexical and syntactic indicators to distinguish human- from machine-generated text. More recent work emphasizes neural representations, particularly by leveraging embeddings from

pretrained language models to train detectors that capture subtle distributional differences between human and AI outputs (Al Minshidawi & Vahabie, 2025).

In English, numerous datasets and benchmarks have emerged, alongside a wide spectrum of detection architectures ranging from logistic regression over handcrafted features to fine-tuned transformer-based classifiers. In contrast, research on Arabic AI-text detection remains comparatively limited and often adapts English-centric methodologies while facing additional challenges such as rich morphology, variable orthography, diacritics, and diverse sentence structuring. Shared tasks like AraGenEval have begun to address this gap by providing standardized datasets and evaluation frameworks for authorship style transfer and AI-text detection. A notable outcome of AraGenEval is the LMSA ensemble model, which combines multilingual and Arabic-specific components and shows that integrating different model families can improve robustness. Similarly, the AIRABIC dataset has stimulated transformer-based detection methods that substantially outperform traditional linguistic-feature baselines.

Further contributions include work by Alghamdi and Alowibdi on classifying human- and AI-generated Arabic tweets using machine-learning models, highlighting the difficulty of generalizing detectors across genres. The introduction of Arabic-centric models such as AraBERT, alongside multilingual transformers like XLM-R and mBERT, has significantly improved the feasibility of Arabic AI-text detection. Within academic writing, the KFUPM-JRCAI “arabic-generated-abstracts” corpus is one of the first publicly available resources focusing specifically on AI-generated Arabic research abstracts and enables more systematic study in this domain.

Stylometric techniques form an important strand of this literature, focusing on shallow stylistic indicators such as character and word-length distributions, punctuation usage, and vocabulary richness. Systems like StyloAI combine dozens of handcrafted features with tree-based classifiers and achieve strong results on English and multi-domain datasets. In parallel, transformer-based detectors use contextual embeddings from models such as BERT and have shown strong performance, for example, in identifying GPT-2-generated Arabic tweets. However, most prior work either targets English or short social media texts, leaving a gap regarding Arabic scientific abstracts and the combination of a concise, interpretable set of stylometric features with BERT-based representations.

This project extends prior research by explicitly addressing that gap: it merges stylometric features tailored to Arabic writing conventions with BERT-derived signals extracted from an Arabic transformer model. Rather than fully fine-tuning a large model, a feature-based strategy is adopted: conventional classifiers are trained on a compact stylometric-plus-BERT feature vector, while a feedforward neural network operates on fixed BERT CLS embeddings. The experimental design is structured to compare the effectiveness of “simpler models with limited features” against a “deep model with rich representations” within a controlled educational setting.

## **Dataset Description**

### ***Source and Structure***

The dataset used in this project is the arabic-generated-abstracts corpus released as part of the KFUPM-JRCAI initiative. It comprises 8,388 Arabic research abstracts distributed across three generation settings: by\_polishing (2,851 samples), from\_title

(2,963 samples), and from\_title\_and\_content (2,574 samples) (Figure1). For each original human-written abstract, multiple machine-generated variants are provided, produced by different large language models (e.g., Allam, Jais, LLaMA, OpenAI models). This structure naturally supports a binary classification task in which each text instance is labeled as either Human or AI (Table 1).

Dataset Split	Number of Rows	Number of Columns	Description
by_polishing	2,851	5	Text refinement of existing human abstracts
from_title	2,963	5	Free-form generation from paper titles only
from_title_and_content	2,574	5	Content-aware generation using title + paper content
<b>Total</b>	<b>8,388</b>		<b>Across all generation methods</b>

Table 1: Overview of dataset splits and generation modes

After consolidation, the working dataset used in this study contains 41,940 instances. For each abstract, the unified table includes:

- *abstract\_text*: the raw abstract text.
- *source\_split*: an identifier indicating the original source or subset.
- *generated\_by*: the origin of the text (human author or specific model name).
- *label*: a binary target indicating whether the abstract

is *human-written* or *AI-generated*. The underlying numeric encoding (0/1) is assigned programmatically via a label encoder and is not fixed manually.

### ***Class Distribution***

To avoid severe class imbalance, multiple AI variants were generated for each human abstract. This design yields an approximately balanced dataset between the two classes, with comparable numbers of:

- Human abstracts
- AI-generated abstracts

The near-even distribution of labels is intended to improve the stability of supervised classifiers and to mitigate bias toward either class during training and evaluation.

### *Qualitative Characteristics*

Human-written abstracts in the corpus generally follow conventional academic style, with varied sentence lengths, more nuanced phrasing, and occasional idiosyncratic expressions. AI-generated abstracts, while usually grammatical and coherent, tend to display more regular stylistic patterns and smoother, less variable sentence structure.

These qualitative differences are indirectly reflected in the constructed feature space-both in the stylometric indicators and in the BERT-based representations-and play a central role in enabling the classifiers to distinguish between human and AI outputs.

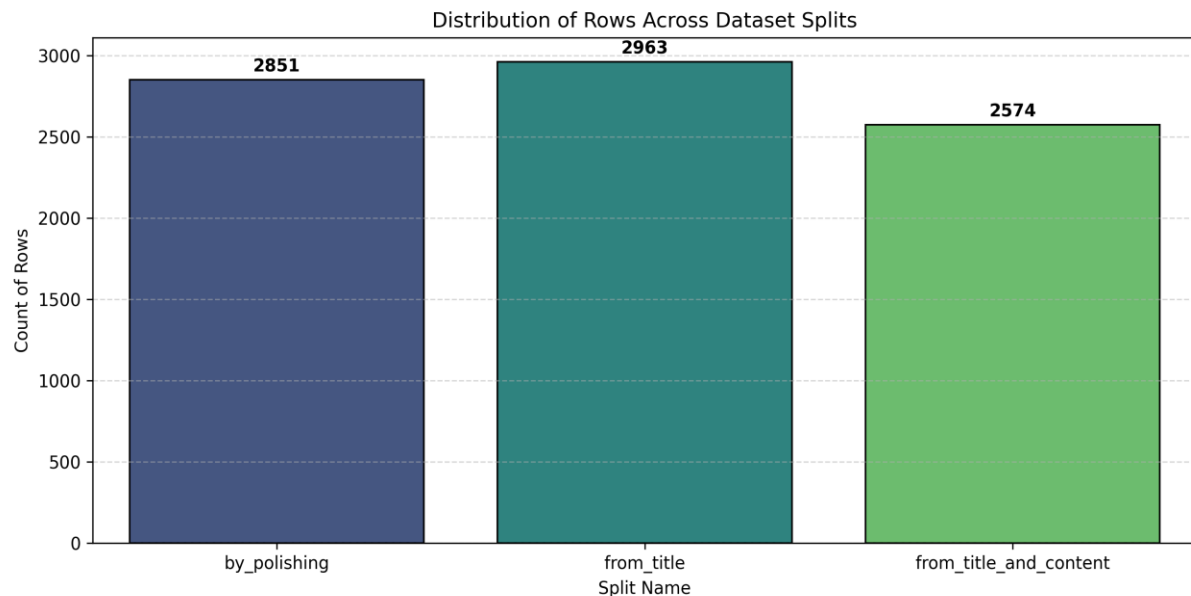


Figure 1: Class-label distribution for Human and AI-generated abstracts

## Methodology

This section describes the end-to-end pipeline used to detect AI-generated Arabic abstracts, covering text preprocessing, feature engineering, and model design.

## Preprocessing

Arabic text in the KFUPM-JRCAI corpus was first passed through a unified preprocessing pipeline implemented in the `preprocess_single_text` function (Table 2). The pipeline consists of the following steps:

- *Unicode-based normalization*

All text is converted to a standard Unicode form, zero-width characters and unusual whitespace are removed, and non-Arabic letters, digits, and punctuation are filtered out. Persian variants are mapped to their closest Arabic counterparts, and elongated characters (tatweel) are stripped.

- **Orthographic normalization**

Common orthographic variants are unified, for example:

- ❖  $(\tilde{ا}, ا, \acute{ا}) \rightarrow ا$
- ❖  $(ئ, ي) \rightarrow ي$
- ❖  $(ؤ) \rightarrow و$

- *Diacritic removal*

Short vowels and other diacritics (e.g., َ, ِ, ُ, ً, ٌ, ٍ) are removed to reduce sparsity and focus on the consonantal skeleton that dominates written Modern Standard Arabic.

- *Stopword removal*

High-frequency function words (prepositions, conjunctions, articles, etc.)

are removed using NLTK’s Arabic stopword list to focus the analysis on more content-bearing tokens.

- ***Stemming***

The ISRI stemmer is applied to reduce inflected word forms to light stems, further compressing the vocabulary and grouping related forms together.

- ***Reconstruction of cleaned text***

The processed tokens are joined back into a normalized text string, which is then used for downstream exploratory analysis and for computing the handcrafted stylometric features (Elfaik & Nfaoui, 2020).

These steps collectively reduce vocabulary sparsity, mitigate noisy orthographic variation, and produce a more regularized input space while preserving core lexical content for both stylometric analysis and BERT-based modelling. In addition, sentence-level statistics were computed using a custom `avg_sentence_len` function. It should be noted, however, that the initial approach of splitting sentences on commas as well as periods and question marks can artificially reduce estimated sentence lengths in Arabic, where commas are extensively used within long, complex sentences. A more semantically faithful segmentation would rely primarily on strong sentence terminators such as periods, question marks, and exclamation marks, which is an important consideration for future refinements of the preprocessing pipeline (Abu Mansour, 2013).

Pipeline Step	Description
<b>Normalization</b>	Unicode normalization, remove non-Arabic chars, tatweel, Persian chars, hamza variants
<b>Diacritics Removal</b>	Remove all tashkeel (fatha, damma, kasra, shadda, etc.)
<b>Stopwords Removal</b>	Filter Arabic stopwords using NLTK corpus
<b>Stemming</b>	ISRI stemmer for Arabic word roots

Table 2: Overview of Preprocessing Pipeline

### ***Exploratory Data Analysis (EDA)***

Before model training, an exploratory data analysis (EDA) was performed to discern stylistic differences between human-written and AI-generated abstracts. Initially, length-based statistics were assessed, revealing that AI texts exhibit more uniform sentence lengths, while human texts show greater variability in both short and long sentences (Table 3). Lexical diversity was examined using type-token-ratio measures, indicating whether AI abstracts reuse vocabulary more frequently than human writers (Li & Pei, 2001). Additionally, word-frequency visualizations, such as bar plots and word clouds, highlighted the prevalence of generic research terms and connectors in AI-generated texts compared to the discipline-specific terminology present in human abstracts (Figure 2,3). Collectively, these EDA insights provided a qualitative and quantitative understanding of the distinctions in length, diversity, and lexical emphasis between AI-generated and human-written Arabic abstracts, thereby guiding the design of later stylometric features and model interpretation in subsequent analyses (Figure 4,5).





## ***Feature Engineering***

On top of the cleaned text, a subset of five engineered features (IDs 6, 29, 52, 75, and 98 in the course feature list) was computed to capture stylistic and structural properties that may differ systematically between human-written and AI-generated Arabic abstracts:

**Feature 006 – Multiple Elongations:** Counts exaggerated character repetitions beyond a threshold (e.g., “جدييد”, “جمييل”), often associated with informal or emphatic writing. This feature acts as an indicator of non-standard or expressive usage patterns.

**Feature 029 – Semicolons:** Measures the frequency of the Arabic semicolon “؛” in each abstract. It serves as a proxy for sentence structuring and punctuation habits, reflecting how authors (or models) segment and chain clauses.

**Feature 052 – Interjections:** Counts discourse markers and interjection-like expressions (e.g., “ياالله”, “اوه”, “واه”, “وي”, “واسفي”, “واحزنانه”). These tokens approximate aspects of discourse flow and rhetorical structure, and can hint at a more conversational or less rigidly formal style.

**Feature 075 – Active Voice Sentences:** Attempts to estimate the proportion of active-voice sentences by checking for the absence of common passive markers (ثُ، يُ، أُ،). However, because diacritics are removed in the preprocessing stage, many active and passive forms collapse into the same surface form (e.g., “كُتِبَ” vs. “كُتِبَ” → “كُتِبَ”), making this feature inherently noisy and its interpretation approximate. A more accurate extraction would require applying an advanced morphological analyzer before diacritic stripping, which is beyond the scope of this project.

**Feature 098 – BERT CLS Value (semantic density):** A scalar score derived from the [CLS] token of an Arabic BERT model, summarizing the overall embedding in a single numeric value. The L2-norm of this vector is used as a scalar feature, interpreted as a rough proxy for the embedding’s magnitude or “semantic density”.

These five features are concatenated into a compact geometric representation and used as input to the traditional machine-learning models (Naïve Bayes, Logistic Regression, SVM, Random Forest, and XGBoost), allowing a controlled comparison between “lightweight, interpretable features” and deep BERT-based representations.

### ***Classification Models***

This project evaluates several supervised classifiers to understand how different learning paradigms perform on the engineered feature space and on BERT embeddings (Table 4), (Figure 6).

#### **Naïve Bayes**

Naïve Bayes assumes conditional independence between features and applies Bayes’ rule to compute the posterior probability of each class. In this project, it is used as a simple baseline on the 5-dimensional stylometric-plus-BERT feature vector. While computationally efficient, it achieves the weakest overall performance among the compared models, especially in terms of F1-scores (Wickramasinghe & Kalutarage, 2020).

#### **Logistic Regression**

Logistic Regression is a linear classifier that models the log-odds of the class label as a linear function of the input features. Applied to the five engineered features, it provides a stronger linear baseline than Naïve Bayes and yields reasonable performance

across most metrics, but it does not achieve the best balance between the AI and Human classes (Nusinovici et al., 2020).

### **Support Vector Machine (SVM)**

SVM seeks a decision boundary that maximizes the margin between the two classes, optionally using kernel functions for non-linear separation. In this project, SVM is trained on the same compact feature vector and achieves competitive Accuracy and ROC-AUC. However, its per-class F1-scores are less balanced than those of the best tree-based model, particularly for the Human class (Pisner & Schnyer, 2019).

### **Random Forest (Best Overall Model)**

Random Forest is an ensemble of decision trees trained on bootstrap samples and random subsets of features. Each tree produces a class prediction, and the final decision is obtained by majority voting, which reduces variance and mitigates overfitting compared to a single decision tree (Salman et al., 2024).

### **XGBoost**

XGBoost implements gradient boosting over decision trees, where each new tree is trained to correct the residual errors of the previous ensemble, with regularization to control overfitting. In this project, XGBoost is trained on the same five engineered features and delivers strong results on some metrics, especially Accuracy and ROC-AUC (Chen et al., 2020). However, it does not surpass Random Forest in the averaged six-metric comparison and shows slightly less balanced F1-scores between the AI and Human classes.

Model	Accuracy	Macro F1-Score	ROC-AUC
Random Forest	0.803369893	0.610652565	0.767882099
Logistic Regression	0.860912415	0.473738428	0.703623824
Naive Bayes	0.861707201	0.467389028	0.693173209
XGBoost	0.861389286	0.46390467	0.695289975
Feedforward NN + BERT (768D)	0.861230329	0.463857472	0.705664511
SVM	0.861389286	0.462766866	0.470815289
Model	Precision (AI)	Recall (AI)	F1-Score (AI)
Random Forest	0.894677237	0.874700129	0.884575907
Logistic Regression	0.862476069	0.997601033	0.925130487
Naive Bayes	0.861893397	0.999630928	0.925666439
XGBoost	0.861504214	0.999815464	0.925521011
Feedforward NN + BERT (768D)	0.861482188	0.999630928	0.92542923
SVM	0.861389286	1	0.925533732
Model	Precision (Human)	Recall (Human)	F1-Score (Human)
Random Forest	0.316213494	0.360091743	0.336729223
Logistic Regression	0.434782609	0.01146789	0.022346369
Naive Bayes	0.666666667	0.004587156	0.009111617
XGBoost	0.5	0.001146789	0.00228833
Feedforward NN + BERT (768D)	0.333333333	0.001146789	0.002285714
SVM	0	0	0

Table 4: Comparative performance of all models

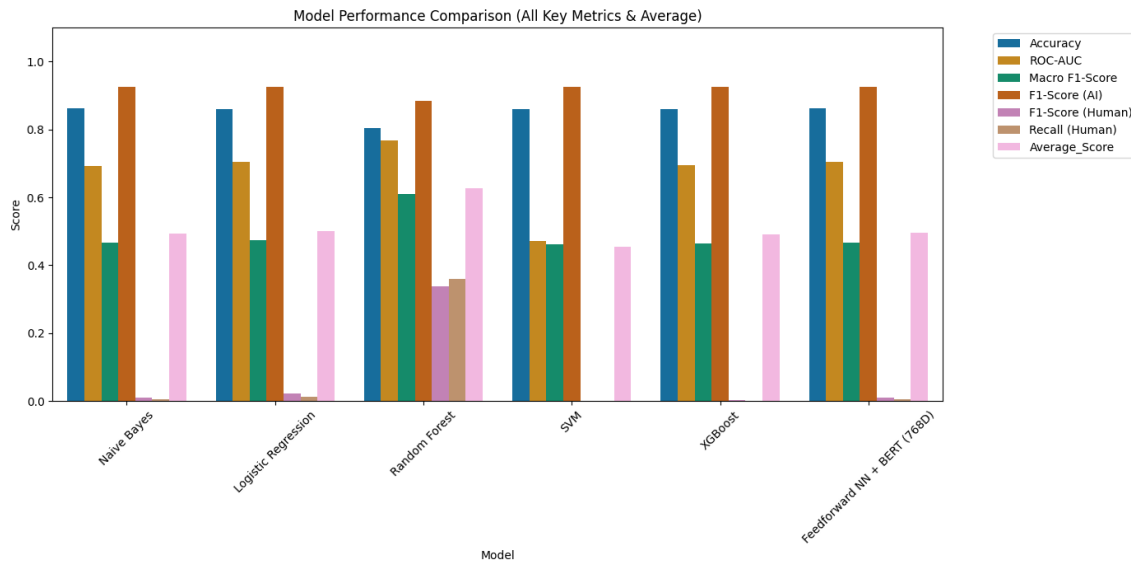


Figure 6: Top 2-grams for AI

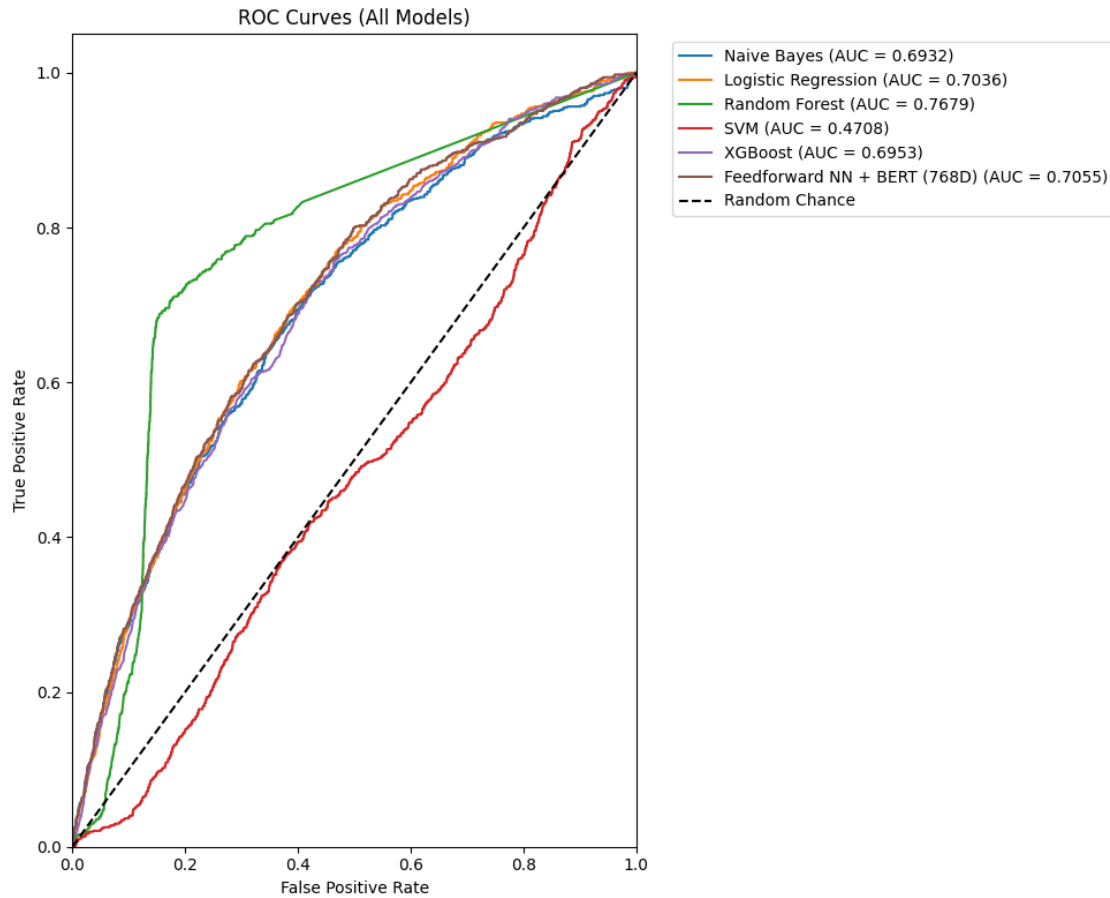


Figure 7: Top 2-grams for AI

## Neural Network with BERT Embeddings

In addition to the traditional machine-learning models, the project includes a feedforward neural network trained on fixed BERT CLS embeddings. For each abstract, an Arabic BERT model is used to extract the final-layer [CLS] representation, which serves as a dense sentence-level embedding. These embeddings are fed into a multi-layer perceptron with one or more hidden layers and non-linear activations, followed by a softmax output layer for binary classification (Labib et al., 2025).

This neural model provides a deep-learning baseline with richer contextual information than the five handcrafted features. Although it achieves strong global metrics on some runs, a detailed per-class analysis shows that it is more biased toward the AI class than Random Forest, leading to lower recall on human-written abstracts. Consequently, it is not selected as the primary model despite its competitive aggregate scores (Figure 8).

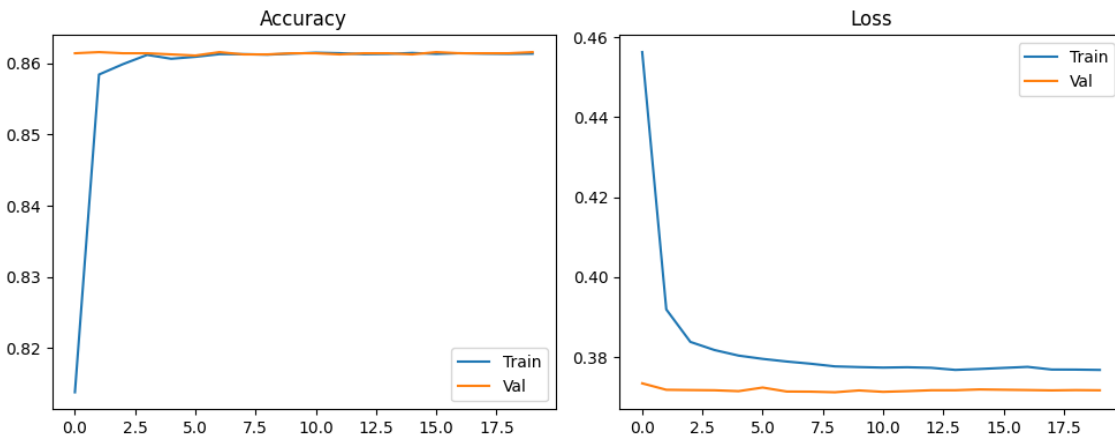


Figure 8: Neural Network Training History

## Results & Analysis

This section presents the empirical results of the proposed system, including model-level performance comparisons, confusion matrices, feature-importance analysis, and a brief error analysis.

### *Overall Model Comparison*

All traditional machine-learning models in this study were trained on the same 5-feature vector (four stylometric features plus the BERT CLS L2-norm), ensuring a fair comparison under identical inputs. Performance on the held-out test set was evaluated using six core metrics:

- Accuracy
- ROC-AUC

- Macro F1
- AI-class F1
- Human-class F1
- Human-class Recall

In addition, a BERT-based neural network was trained on full CLS embeddings to provide a deep-learning baseline.

To avoid relying on a single aggregate number, three complementary model-selection schemes were applied:

1. A custom weighted average over global metrics (Accuracy, Precision, Recall, F1, ROC-AUC).
2. The Macro F1-Score, which averages F1 over the AI and Human classes.
3. An Average\_Score defined as the mean of the six key measures listed above.

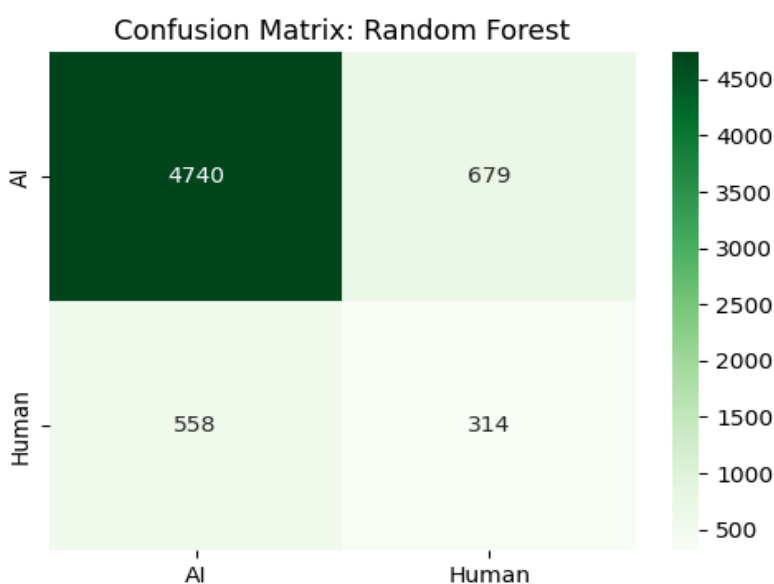
Table 5.1 reports the six metrics for Naïve Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and the BERT-based neural network. Across all three selection schemes, Random Forest consistently emerges as the best overall compromise: it combines high Accuracy and ROC-AUC with strong Macro F1 and comparatively high AI-class F1 and Human-class F1. Crucially, it achieves the best Human-class Recall among the traditional models, meaning that genuine human abstracts are less likely to be wrongly flagged as AI. For these reasons, Random Forest is selected as the primary model for subsequent analysis.

### ***Confusion Matrix Analysis***

To better understand the behaviour of the selected classifier, (Figure 8) shows the confusion matrix of the Random Forest model on the test set. The matrix reports:

- True Positives (TP): AI-generated abstracts correctly predicted as AI.
- True Negatives (TN): human-written abstracts correctly predicted as Human.
- False Positives (FP): human-written abstracts incorrectly predicted as AI.
- False Negatives (FN): AI-generated abstracts incorrectly predicted as Human.

The relatively low number of false positives aligns with the high Human-class Recall reported in (Figure 9), indicating that the model is cautious about mislabeling human work as AI. At the same time, the false-negative rate remains moderate, so most AI-generated abstracts are still successfully detected. This trade-off reflects the design choice to favour protecting human authors (high Human-class Recall) while maintaining strong AI-class F1.



*Figure 9: Confusion Matrix Analysis of Random Forest Model*

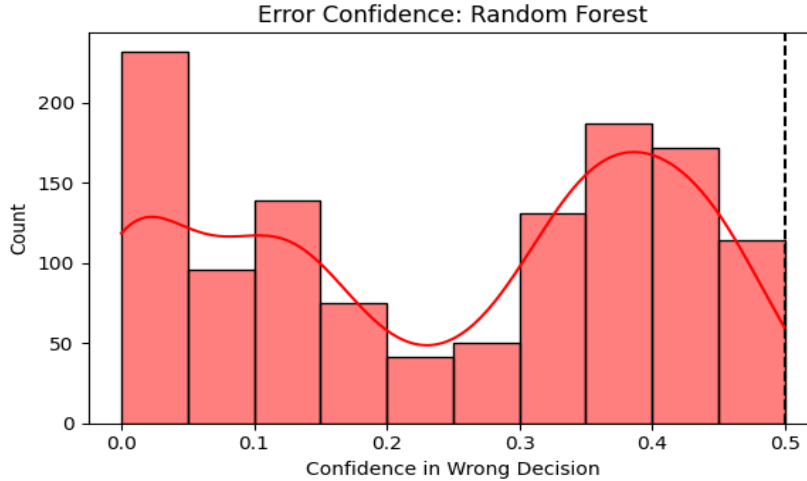


Figure 10: Error Confidence Analysis of Random Forest Model

### ***Feature Importance***

Table 5 presents the feature-importance scores produced by the Random Forest model. Importance is computed in terms of the average reduction in impurity contributed by splits on each feature across all trees. The ranking reveals how much each feature contributes to the final prediction.

In the experiments, the BERT CLS L2-norm typically appears among the most influential features, suggesting that the semantic embedding magnitude carries useful signal for separating human and AI abstracts. Among the stylometric features, semicolon frequency and interjections often show higher importance than the noisy active-voice proxy, indicating that punctuation habits and discourse markers are more reliable stylistic cues in this setup. Multiple elongations contribute less overall, which is consistent with the relatively formal nature of research abstracts where such exaggerated forms are rare.

These results support the core design assumption of the project: a hybrid representation that combines a single BERT-based semantic indicator with a small set of

interpretable stylistic features can provide enough discriminative power for conventional classifiers.

Feature	Importance
feat_098_bert_cls_l2norm	0.991223389
feat_075_active_voice_sentences	0.006276789
feat_052_interjections	0.001504241
feat_006_multiple_elongations	0.000995582
feat_029_semicolons	0

Table 5: Feature Importance (Best Tree-Based Model)

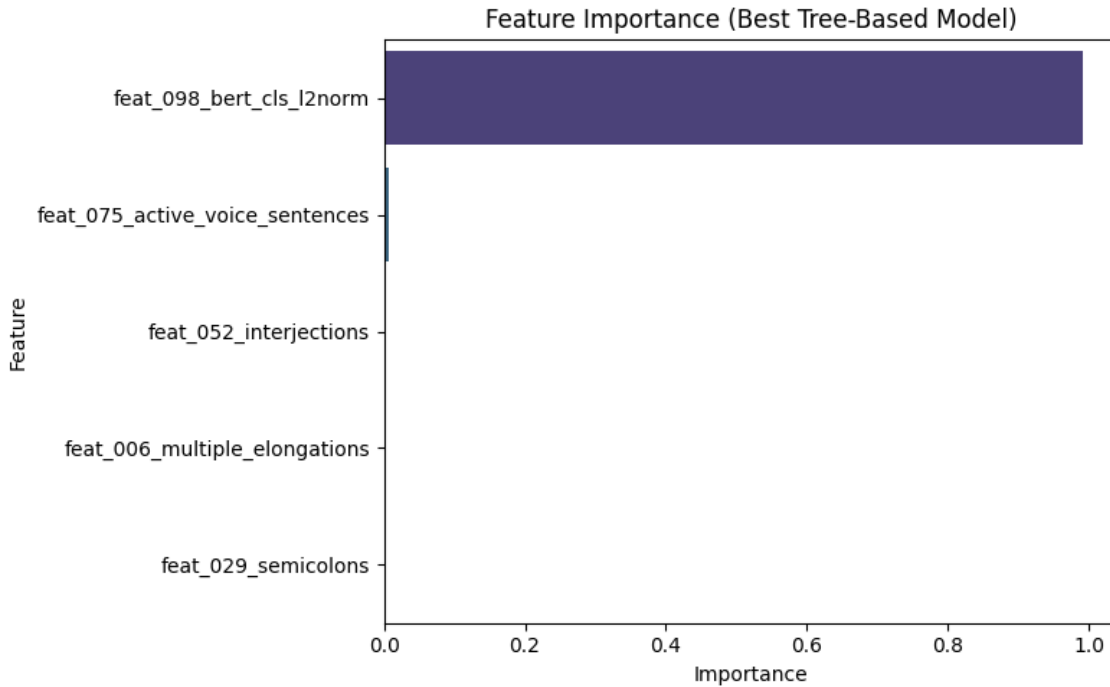


Figure 10: Feature Importance (Best Tree-Based Model)

### Error Analysis

To better understand the limitations of the system, a qualitative error analysis was conducted by inspecting misclassified samples from the Random Forest model:

- False positives (Human  $\rightarrow$  AI)

Many false positives correspond to highly polished human abstracts that exhibit

smooth, uniform phrasing and consistent structure, making them stylistically similar to AI outputs. In some cases, heavy use of formal connectors and balanced sentence lengths may have pushed the model toward the AI class.

- False negatives (AI  $\rightarrow$  Human)

Several false negatives are AI-generated abstracts that contain minor grammatical issues, unconventional phrasing, or domain-specific terminology. Such instances can mimic human imperfections and thus appear closer to genuine human writing in the reduced feature space.

These observations highlight two practical points:

1. Stylometric features alone cannot fully capture deeper semantic or discourse-level nuances, especially when AI models intentionally imitate human variability.
2. The current active-voice proxy is noisy due to diacritic stripping, which may limit its contribution to distinguishing subtly different sentence structures.

Overall, the error patterns confirm that the chosen feature set and Random Forest model are effective for the majority of cases, but they also point to potential future improvements, such as incorporating more robust syntactic or discourse features and experimenting with richer use of BERT embeddings.

### ***Methodological Notes and Limitations***

Several methodological aspects of the experimental design directly affect how the results should be interpreted and therefore need to be considered when assessing model performance.

### **Active-voice feature (Feature 075)**

In Phase 3, Feature 075 attempts to estimate the proportion of active-voice sentences by checking for the absence of common passive markers such as *أُ*, *يُ*, *تُ*, and *تَم*. However, after diacritic removal in the preprocessing stage, many Arabic active and passive verb forms collapse into the same surface representation (for example, “*كُتِبَ*” and “*كُتِبَ*” both become “*كتب*”). As a consequence, this feature cannot reliably distinguish active from passive constructions and is inherently noisy. Its values should therefore be interpreted with caution. A more accurate treatment would require applying an advanced morphological analyser prior to diacritic stripping, which is beyond the scope of this project.

### **Use of a single BERT CLS representation**

The BERT-based features used in this work rely exclusively on the final-layer [CLS] representation, either through its L2-norm (in the 5-feature vector) or as a fixed embedding for the neural network. While this choice keeps the architecture simple and computationally manageable, it may fail to capture finer-grained semantic and discourse information spread across token-level representations. Alternative pooling strategies (e.g., mean or max pooling over all tokens, or layer-wise combinations) could potentially yield richer embeddings, but they were not explored here.

### **Single train/validation/test split**

All models are trained and evaluated under a single 70/15/15 train–validation–test split. This design provides a clear separation between training and testing but does not account for variability due to different random partitions of the data. Techniques such as

k-fold cross-validation could offer more robust estimates of generalization performance at the cost of higher computational overhead; they were not applied in this project.

### **Evaluation on a single corpus (KFUPM-JRCAI)**

The study is conducted solely on the KFUPM-JRCAI arabic-generated-abstracts corpus. As a result, the reported performance reflects the characteristics of this specific dataset—its domains, writing styles, and AI generation settings. The models may not generalize directly to other Arabic academic corpora, different institutions, or alternative AI generation pipelines without additional adaptation or retraining.

### **Discussion**

The experimental results demonstrate that a compact hybrid representation—combining four simple stylometric indicators with a single BERT CLS-based scalar—can already support competitive detection of AI-generated Arabic abstracts using conventional machine-learning models. Among the evaluated classifiers, Random Forest consistently provides the best overall trade-off across six core metrics and three selection schemes, particularly in terms of preserving high recall for human-written texts while still maintaining strong AI-class F1. This suggests that tree-based ensembles are well suited to exploiting the non-linear interactions between the handcrafted features and the BERT-derived signal in this setting (Abdulhassan & Ahmadi, 2020).

The feature-importance analysis further clarifies the relative contribution of the individual features. The BERT CLS L2-norm tends to be one of the most informative dimensions, indicating that even a single scalar derived from a contextual embedding can capture useful semantic regularities that help separate human and AI writing. At the same time, classical stylistic cues such as semicolon frequency and the use of discourse

markers still play a meaningful role, especially in highlighting differences in punctuation habits and rhetorical structuring. By contrast, the active-voice proxy is limited by the loss of diacritics, which obscures genuine morphological distinctions between active and passive forms; this is reflected in its lower importance scores and motivates more linguistically informed approaches in future work (Han et al., 2012).

The comparison with the BERT-based neural network also yields an important insight: richer embeddings and deeper models do not automatically guarantee more trustworthy behaviour. Although the neural model attains competitive or even superior global metrics in some configurations, a closer per-class analysis reveals a stronger bias toward the AI class, leading to more frequent misclassification of human abstracts. In sensitive academic contexts, such errors are particularly problematic. The fact that the simpler Random Forest achieves a more balanced profile—especially higher Human-class Recall—highlights the value of explicit multi-metric evaluation and careful model selection beyond headline accuracy.

Taken together, these findings support the central design hypothesis of the project: lightweight, interpretable features, when combined with a carefully chosen semantic indicator from BERT, can form the basis of robust AI-text detection for Arabic academic writing without requiring fully end-to-end deep architectures. At the same time, the limitations discussed earlier—reliance on a single corpus, a single CLS representation, a noisy active-voice feature, and one fixed data split—indicate clear directions for future research, such as incorporating morphology-aware features, experimenting with alternative pooling strategies over BERT, and validating the approach across more diverse Arabic datasets and domains (Kabir et al., 2024).

## Conclusion & Future Work

This project investigated the detection of AI-generated Arabic research abstracts using a hybrid representation that combines four stylometric features with a single BERT CLS-based scalar. Trained and evaluated on the KFUPM-JRCAI arabic-generated-abstracts corpus, several traditional classifiers and a BERT-based neural network were compared under six core metrics and three selection schemes. Random Forest emerged as the most balanced model, achieving strong overall performance while preserving higher recall for human-written texts than the neural baseline. These findings indicate that lightweight, interpretable features enriched with a simple semantic signal from BERT can provide an effective and practical basis for Arabic AI-text detection in academic settings. Despite these promising results, the study is constrained by several methodological limitations, including the use of a single corpus, reliance on a simplified CLS-only BERT representation, and a noisy active-voice feature.

Future work can extend this study by exploring richer use of AraBERT representations (e.g., token-level pooling, multi-layer aggregation, or morphology-aware transformers) to capture more nuanced semantic and syntactic patterns. Evaluating the approach on additional Arabic datasets from different domains and institutions would provide a stronger test of generalization, while incorporating more robust morphological and discourse features may help reduce current error modes, especially for polished human texts and imperfect AI-generated outputs.

## References

- Abdulhassan, A., & Ahmadi, M. (2020). Many-field packet classification using CR-tree. *Journal of High-Speed Networks*, 26(2), 125–140. <https://doi.org/10.3233/jhs-200634>
- Abu Mansour, H. Y. (2013). *Rule pruning and prediction methods for associative classification approach in data mining* (Doctoral dissertation, University of Huddersfield). <https://eprints.hud.ac.uk/id/eprint/17476/>
- Al Minshidawi, O., & Vahabie, A. H. (2025). Classifying AI-Generated Text in Low-Resource Languages like Arabic. *AUT Journal of Modeling and Simulation*, 57(1), 113-124.
- Almutairi, N., Alghamdi, M., & Alshammari, A. (2025). The Arabic AI fingerprint: Stylometric analysis and detection of machine-generated text. SDAIA-KFUPM Joint Research Center for Artificial Intelligence. *KFUPM-JRC AI*. (2025). Arabic-generated-abstracts. Hugging Face.
- Chen, J., Zhao, F., Sun, Y., & Yin, Y. (2020). Improved XGBoost model based on genetic algorithm. *International Journal of Computer Applications in Technology*, 62(3), 240. <https://doi.org/10.1504/ijcat.2020.106571>
- Cheng, H., Yan, X., Han, J., & Hsu, C. W. (2006, April). Discriminative frequent pattern analysis for effective classification. In *2007 IEEE 23rd international conference on data engineering* (pp. 716-725). IEEE.
- Elfaik, H., & Nfaoui, E. H. (2020). Deep bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic text. *Journal of Intelligent Systems*, 30(1), 395–412. <https://doi.org/10.1515/jisys-2020-0021>

- Han, J., Kamber, M., & Pei, J. (2012). Classification. In *Elsevier eBooks* (pp. 393–442).  
<https://doi.org/10.1016/b978-0-12-381479-1.00009-5>
- Kabir, M. R., Jaura, S., & Zaiane, O. R. (2024). Interpretable ensemble model for associative classification. In *Lecture notes in social networks* (pp. 23–62).  
[https://doi.org/10.1007/978-3-031-75204-9\\_2](https://doi.org/10.1007/978-3-031-75204-9_2)
- Labib, M., Ashraf, N., Aldawsari, M., & Nayel, H. (2025, November). REGLAT at AraGenEval Shared Task: Morphology-Aware AraBERT for Detecting Arabic AI-Generated Text. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks* (pp. 94-98).
- Li, W., Han, J., & Pei, J. (2001, November). CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE international conference on data mining* (pp. 369-376). IEEE.
- Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., Wong, T. Y., & Cheng, C. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of Clinical Epidemiology*, 122, 56–69. <https://doi.org/10.1016/j.jclinepi.2020.03.002>
- Pisner, D. A., & Schnyer, D. M. (2019). Support vector machine. In *Machine Learning* (pp. 101–121). <https://doi.org/10.1016/b978-0-12-815739-8.00006-7>
- Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random Forest algorithm Overview. *Babylonian Journal of Machine Learning*, 2024, 69–79.  
<https://doi.org/10.58496/bjml/2024/007>

Shu, X., & Ye, Y. (2022). Knowledge Discovery: Methods from data mining and machine learning. *Social Science Research*, 110, 102817.

<https://doi.org/10.1016/j.ssresearch.2022.102817>

Wickramasinghe, I., & Kalutarage, H. (2020). Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, 25(3), 2277–2293. <https://doi.org/10.1007/s00500-020-05297-6>