Human and AI Arabic Text Detection

Introduction

This study presents a framework for detecting AI-generated Arabic text using machine learning. Results show that classical models trained with TF-IDF and Arabic-specific features outperform a neural network based on sentence embeddings, highlighting the importance of feature engineering. processing.

Dataset

Corpus Composition

The dataset comprises parallel collections of academic abstracts:

- **Human-written abstracts:** Authentic scholarly content from Arabic academic publications
- **AI-generated abstracts:** Synthetic text produced by large language models
- **Total samples:** 41,940 labeled instances

This balanced corpus enables robust model training and comprehensive evaluation across diverse writing styles and domains.

Feature Engineering

Hapax Dislegomena Ratio

Measures vocabulary repetition by counting words appearing exactly twice, revealing lexical reuse patterns and vocabulary diversity characteristics unique to human or AI writing.

Total Paragraphs (P)

Captures structural organization by quantifying paragraph divisions, reflecting compositional strategies and document architecture patterns in the text.

Feature Engineering

Average Words per Paragraph

Reflects paragraph length distribution and writing style consistency, indicating whether content follows natural human pacing or algorithmic uniformity.

Type-Token Ratio (TTR)

Measures lexical diversity by comparing unique words to total word count:

TTR = Unique Words / Total Words

Higher ratios indicate richer vocabulary and more varied expression patterns.

Feature Engineering

Tanween Frequency

Tanween—the characteristic nunation marks in Arabic (ً ٍ ٌ)—serves as a critical linguistic indicator of grammatical naturalness and morphological authenticity.

Metric: Tanween Characters / Total Tokens

This feature captures morphological richness unique to proper Arabic syntax. AI models often exhibit distinctive tanween usage patterns that differ from human writers, making this an effective discriminative feature for detection tasks.

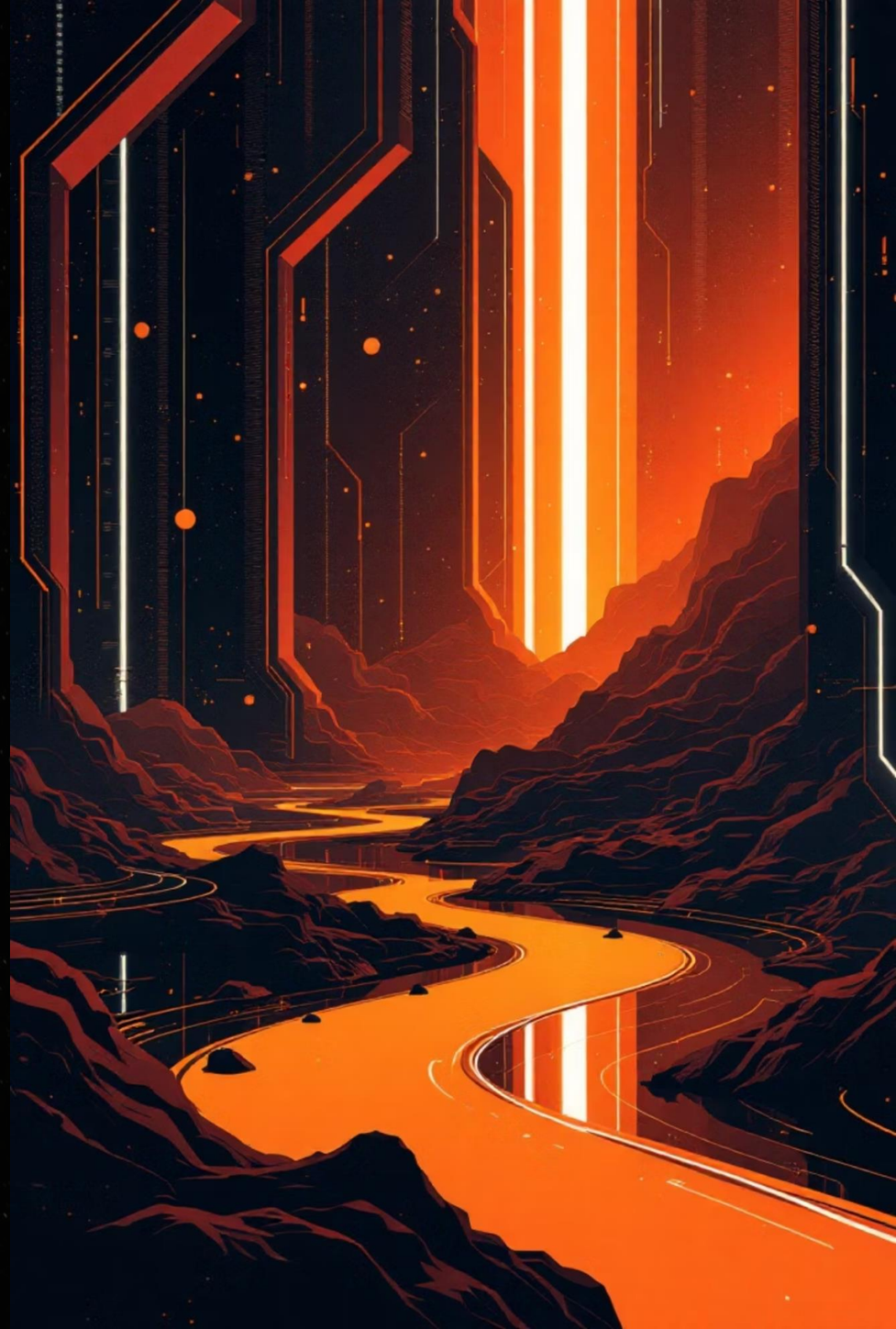
Classical Machine Learning Models

Logistic Regression

Baseline Model

Logistic Regression serves as our foundational baseline due to its computational efficiency and proven performance on high-dimensional sparse features.

Utilizing TF-IDF vectorization combined with handcrafted linguistic features, this model establishes a linear decision boundary for classification. Despite its simplicity, it provides interpretable coefficients and competitive accuracy, offering valuable insights into feature importance for distinguishing AI-generated from human-written Arabic text.



Classical Machine Learning Models

Support Vector Machine

SVM excels at handling high-dimensional feature spaces by constructing optimal hyperplanes that maximize the margin between classes.

By projecting TF-IDF and linguistic features into higher dimensions, SVM identifies complex decision boundaries that effectively separate AI-generated and human-written Arabic text. The kernel trick enables non-linear classification while maintaining computational tractability, resulting in strong generalization performance across diverse text samples.

Classical Machine Learning Models

Random Forest Classifier

Random Forest employs ensemble learning by aggregating predictions from multiple decision trees, each trained on bootstrapped subsets of the data with random feature selection.

This ensemble approach provides several advantages:

- **Robustness:** Reduces overfitting through averaging
- **Non-linearity:** Captures complex feature interactions
- **Feature importance:** Identifies most discriminative attributes

By combining TF-IDF representations with handcrafted linguistic features, Random Forest achieves high classification accuracy while maintaining interpretability.

Classical Machine Learning Models

XGBoost Classifier

XGBoost implements gradient boosting with advanced regularization techniques, iteratively constructing an ensemble of weak learners that focus on previously misclassified samples.

Key strengths:

- Models complex feature interactions and non-linear patterns
- Handles class imbalance through weighted learning
- Provides built-in feature importance rankings
- Achieves state-of-the-art performance through adaptive learning rates

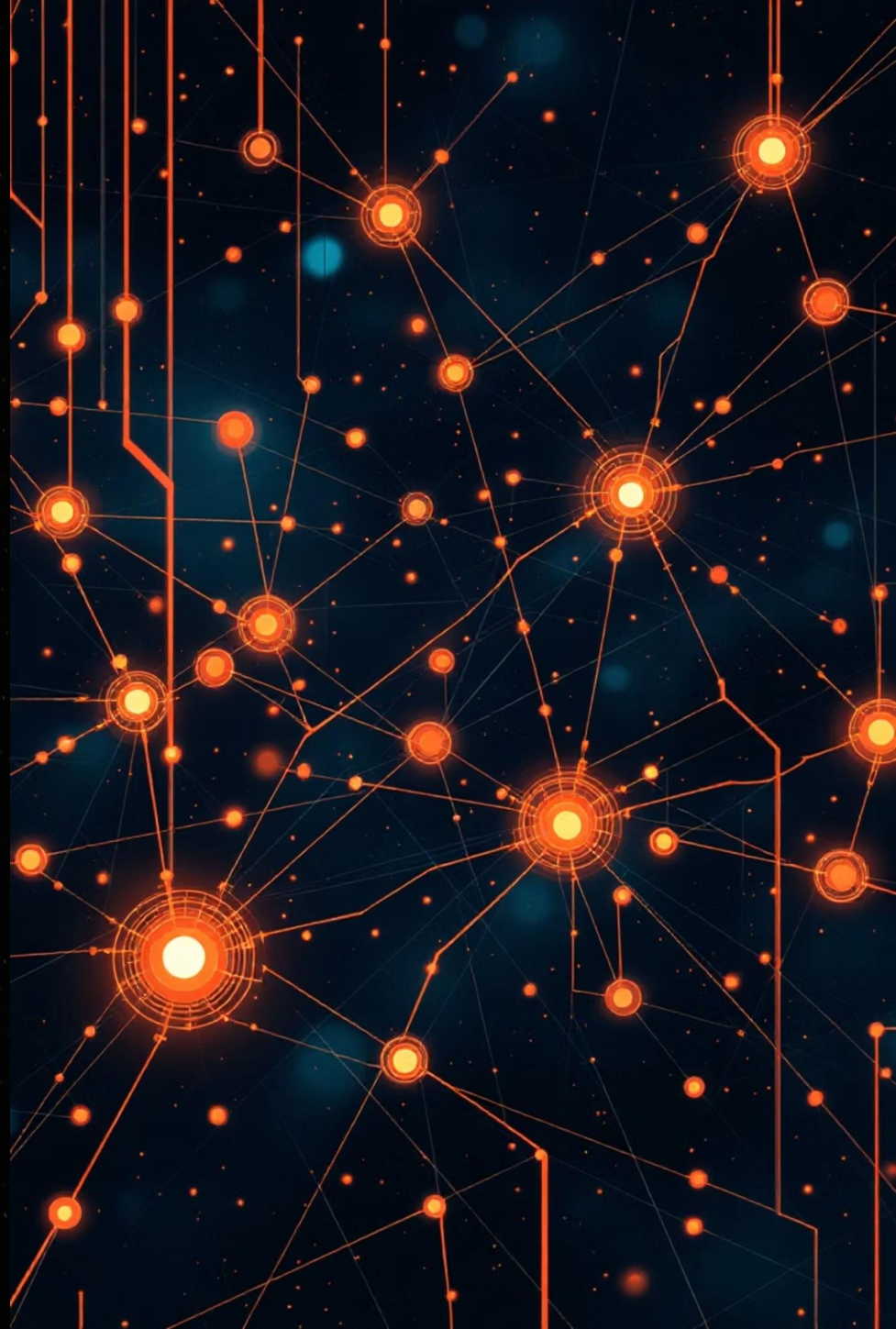
XGBoost emerged as one of the strongest performing models in our comparative study, demonstrating exceptional ability to distinguish subtle differences between human and AI-generated Arabic text.

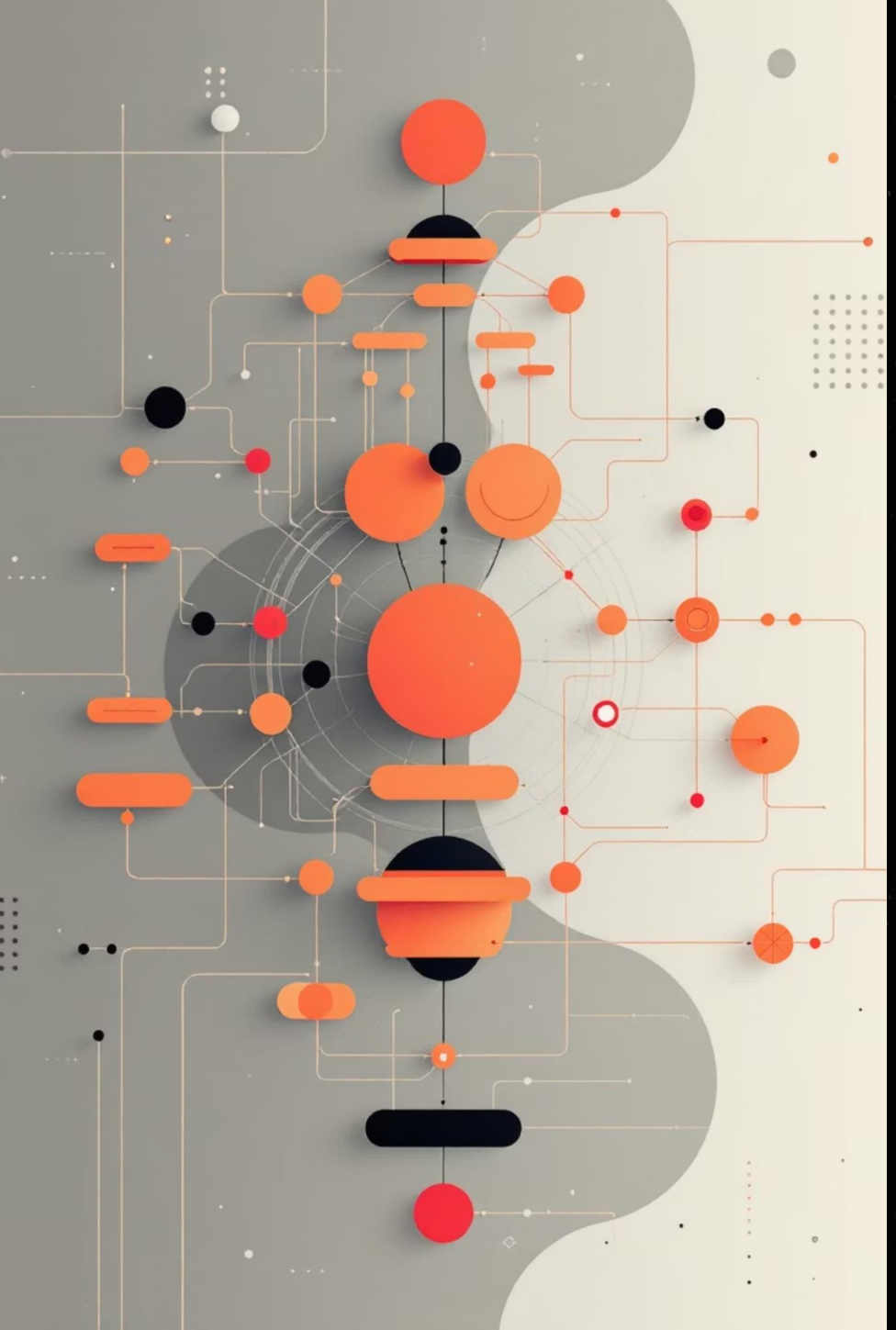
Feedforward Neural Network for Arabic Text Classification

Embedding Strategy

Pre-trained sentence-transformer embeddings serve as fixed semantic representations, capturing rich contextual meaning from Arabic text. These embeddings act as a sophisticated feature extractor, transforming sentences into dense vector representations that preserve linguistic nuances and semantic relationships inherent in the Arabic language.

The resulting high-dimensional vectors are then fed into a feedforward neural network architecture designed specifically for binary classification tasks, distinguishing between AI-generated and human-written Arabic content.





FFNN Architecture Design

01

Input Layer

Receives sentence-level embeddings as fixed-length semantic vectors from pre-trained transformer models

02

Dense Layers

Fully connected layers with ReLU activation functions enable non-linear transformations and feature learning

03

Regularization

Dropout layers strategically placed between dense layers to prevent overfitting and improve generalization

04

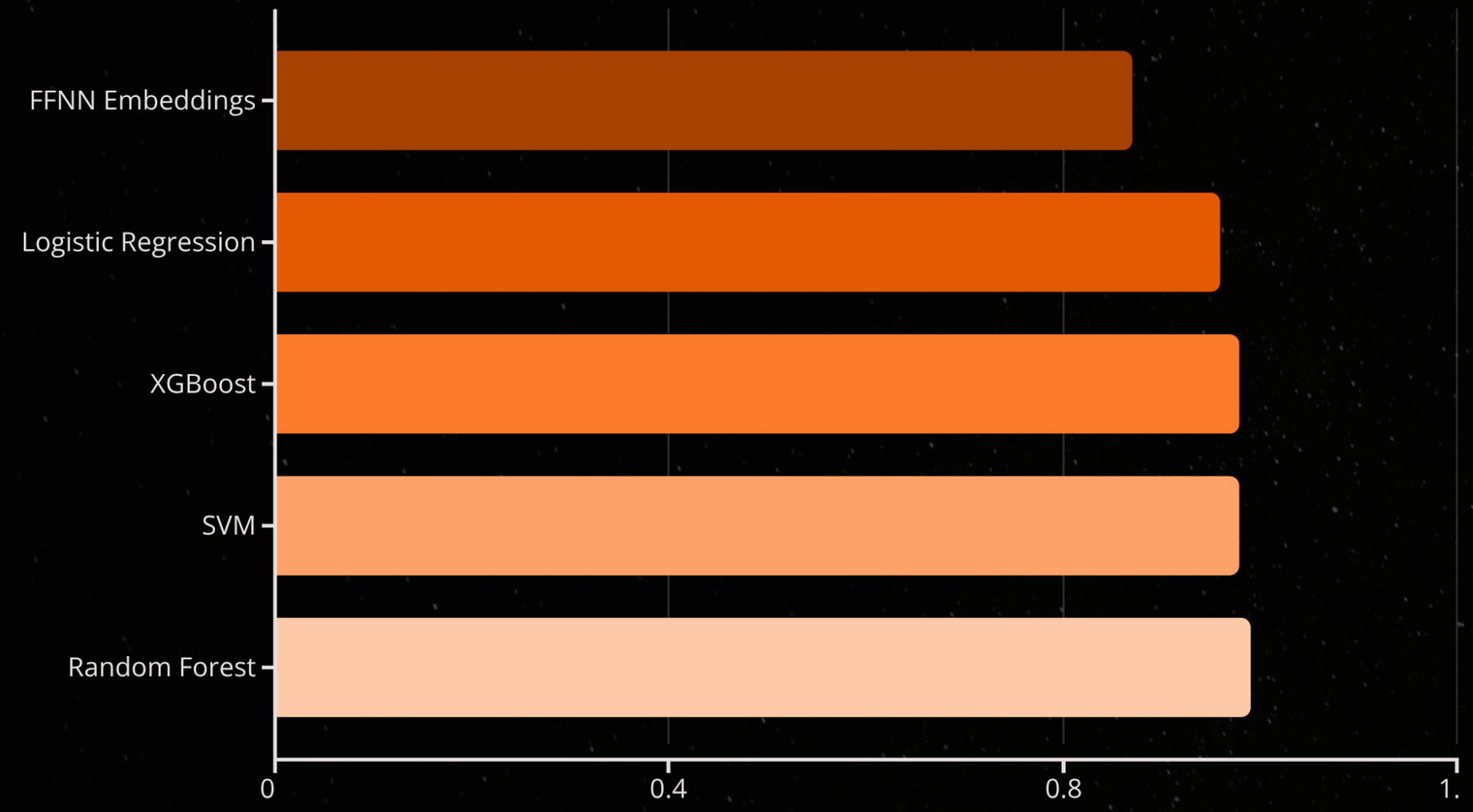
Output Layer

Sigmoid activation produces probability scores for binary classification: AI-generated versus human-written text

Comparative Model Performance

Evaluation across multiple machine learning approaches reveals significant performance variations. Classical methods with handcrafted features demonstrate superior accuracy compared to the embedding-based neural network approach.

FFNN Embeddings: 87% | Logistic Regression: 96% | XGBoost: 98% | SVM: 98% | Random Forest: 99%



Random Forest achieved the highest accuracy at 99%, while the FFNN with fixed embeddings reached approximately 87% accuracy, suggesting that linguistic feature engineering provides stronger discriminative signals for this task.

Key Conclusions

Framework Development

This research established a comprehensive machine learning framework specifically designed for detecting AI-generated Arabic text, incorporating Arabic-specific preprocessing techniques and domain-adapted feature engineering strategies.

Performance Insights

Classical machine learning models leveraging TF-IDF vectorization combined with handcrafted linguistic features significantly outperformed the feedforward neural network approach based on fixed sentence embeddings, achieving accuracy rates up to 99%.

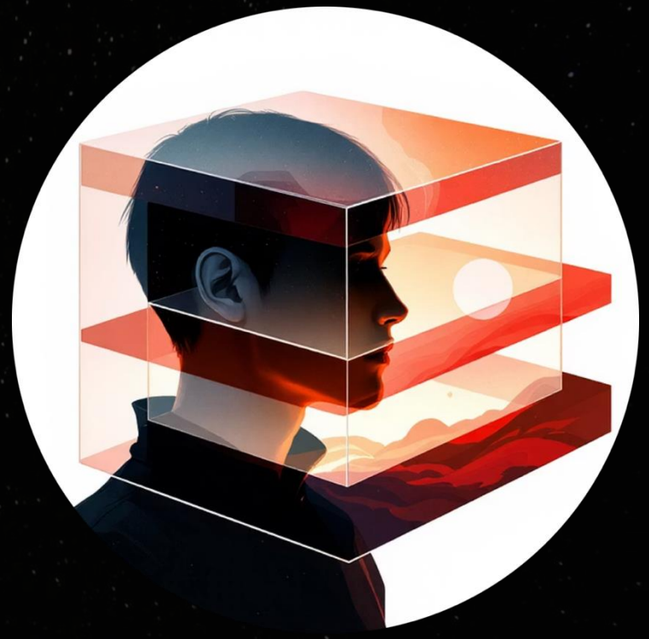
Future Research Directions

Hybrid Feature Approaches

Investigate architectures that combine deep embedding representations with handcrafted linguistic features to leverage both semantic depth and explicit grammatical patterns

Dialectal Coverage

Expand datasets to include regional Arabic dialects and colloquial variations, improving model robustness across the diverse Arabic linguistic landscape



Ensemble & Explainability

Apply ensemble learning techniques and interpretability methods to enhance detection reliability and provide insights into decision-making processes against advanced generative models