Ingénierie des Systèmes d'Information
Vol. 30, No. 12, December, 2025, pp. 3103-3116
Journal homepage: http://iieta.org/journals/isi

Interpretable Multi-Label Classification of Human- and Large Language Model-Generated Texts Using Transformer Embeddings and Explainable Artificial Intelligence

Zahraa J. Mohammed Ali*, Suhad A. Yousif

Department of Computer Science, College of Science, Al-Nahrain University, Baghdad 64074, Iraq

Corresponding Author Email: zahraa.msco23@ced.nahrainuniv.edu.iq

https://doi.org/10.18280/isi.301203

Received: 22 August 2025
Revised: 24 November 2025
Accepted: 8 December 2025
Available online: 31 December 2025

Keywords:
explainable artificial intelligence, LIME, Large Language Model-generated text detection, model attribution, multi-label classification, SHAP, transformer embeddings, XGBoost

ABSTRACT

With the rapid growth of Large Language Models (LLMs), there is an increasing demand for robust algorithms that can differentiate human-written content from text generated by diverse LLM architectures. Current detectors are mostly binary classifiers and lack interpretability, so they cannot be specified to the model responsible for text generation. In this work, we propose an explanation-aware multi-class prediction framework to distinguish human writing from various LLM sources, providing transparent explanations for end-to-end model predictions that rely on an interpretable attribution mechanism. The proposed methodology aggregates four transformer-based embeddings (D-BERT, E5-base, MPNet, and General Text Embeddings – Large (GTE-Large)) using an XGBoost classifier with SHAP and LIME to provide post-hoc interpretability at the feature and token levels. Experiments were conducted on a dataset of 147,834 samples from 17 LLM families and human authors. The E5-based embeddings, combined with XGBoost, achieved the best performance, with an accuracy of 0.89 and an F1-score of 0.88. Explainability analysis also identified distinct language signatures among LLMs, indicating that the model can attribute authorship beyond the human–AI divide. This work contributes a transparent and scalable solution for this field, with practical relevance in academic integrity, misinformation detection/tracing, and digital content evidence analysis.

1. INTRODUCTION

Text categorization is an important NLP approach that classifies text into predefined categories or classes. It is commonly used for tasks such as sentiment analysis, spam filtering, and document classification [1-4]. A more recent and challenging text classification application is content origin detection, which refers to automatically determining whether an article was written by a person or generated by a machine. In recent times, Large Language Models (LLMs) such as BotGPT, ChatGPT, LLaMA, Claude, GLM, and Bloom have made it possible to generate human-like level text.

This proliferation poses practical challenges in distinguishing human-written work from AI-generated content in areas such as education, journalism, and law. One of the most challenging aspects is enabling machines to understand and interpret human language directly from text using machine learning (ML) [5] and deep learning techniques for textual feature extraction. Uniqueness and transparency of text message content are key necessary conditions for maintaining trust and accountability in online relationships [6]. Traditional classifiers are routinely designed as "black boxes" through which we can hardly penetrate their decision-making mechanisms. This ambiguity can lead to a lack of confidence and understanding, especially in sensitive model applications. It is suggested that NetSHAP+LIME will be the describable AI, explainable artificial intelligence (XAI) approach, as SHAP and LIME. These approaches are used to interpret the predictions of a model to enable users to understand what the model does, thereby enabling trust in AI systems [7, 8]. Although modern text classification models have shown substantial progress, most existing studies focus on binary classification settings and do not adequately address the challenges posed by AI-generated texts. Moreover, current approaches rarely provide comprehensive and detailed explanations of model behavior. To bridge this gap, we propose a multi-class classification framework that distinguishes between human-written text and text generated by multiple LLMs.

SHAP and LIME are applied through XAI as Post-Hoc explanations to identify which features and text extracts are important to the model. By doing so, we can make the model transparent without compromising its predictive performance, while remaining mindful of interpretability [9]. To verify this, a balanced training set of 147,834 texts was used, extracted from the Kaggle "Human vs. LLM Text Corpus". A balanced distribution across the 18 classes was ensured by a multi-stage sampling approach that combined random oversampling and stratified sampling, enabling unbiased multi-class classification while optimizing efficiently.

Sophisticated embedding methods, such as DistilBERT, E5-base, MPNet, and General Text Embeddings – Large (GTE-Large), were employed to represent fine-grained semantic variation in texts. These embeddings were then

combined with the XGBoost algorithm to perform the classification. Given that performances, interpretability, and inferential behavior of models may vary across these various forms of embeddings, we in this work perform a thorough comparison on a global scale regarding how each embedding possibility contributes to model classification of human versus generated text by aiming to understand the potential and limitations of each such embedding approach. Our work extends previous results that justify the application of explainable AI techniques for trustworthy and transparent categorization frameworks [10].

The remainder of this paper is organized as follows:

Section 2 Related Work: This section reviews previous work on AI content identification, including both traditional and transformer-based methods, and highlights existing limitations and challenges.

Section 3 Methodology: describes the modus operandi followed throughout the course of this work, from data gathering and preprocessing (text cleaning, label filtering, and data balancing) to testing the generated models. Then, it explains the feature extraction step using various transformer-based methods, data partitioning, and XGBoost model fitting and evaluation. To clarify the whole process, we present a flow diagram.

Section 4 Results and Discussion present the experimental results, performance comparisons across different embedding-based models, and interpretability insights via visual explanations with SHAP and LIME.

Section 5: Conclusion summarizes the contributions and lays the path for building an even more robust, scalable, and interpretable automated text classification framework.

2. RELATED WORK

The rapid proliferation of LLMs, such as GPT, has amplified the need for explainable classifiers that can reliably distinguish between human-authored and machine-generated text. Early in 2016 - 2017 [11, 12], which introduced interpretable methodological foundations and global feature attribution. These tools have since been widely adopted for text classification, enabling inspection of word- or feature-level contributions to a classifier's decision — a critical requirement in the high-stakes setting of AI-generated content detection. In recent years, many XAI approaches have been applied to text classification tasks, particularly for detecting AI-generated text and addressing multi-label classification. A pioneering work by Khosravi et al. [13] addressed the use of XAI in educational systems, with an emphasis on transparency and trustworthiness for learners, using interpretability techniques such as LIME and SHAP. Although these authors did not use the model itself, and no standardized evaluation metrics were proposed or the LCE test case used, this study laid a foundation for understanding LCE design that has been applied in later research. Extending this in 2023, Weng et al. [14] studied the identification of AI-written scientific content, with an emphasis on joint human–machine authorship articles. They rely on a visualization-based explanation in their model to improve interpretability; however, they did not report conventional performance metrics such as accuracy or F1 score. In the same year, Shah et al. [15] introduced a stylistic-feature framework enriched with XAI for identifying AI-generated texts, focusing more on qualitative generalisability analysis than on numerical benchmarks. Similarly, Hajialigol et al. [16] developed XAI-CLASS for a weakly supervised, low-resource setting and utilized LIME and SHAP as post-hoc rationale generation processes. No quantitative results were presented; however, the method emphasized the value of transparency in the face of limited data. In 2024, empirical studies began to consolidate the link between XAI and quantitatively observable classification performance. Zahoor et al. [17] trained XAI-enhanced classifiers on small datasets and reported an accuracy of 85%, concluding that SHAP is more stable than LIME for feature attribution. de Arriba-Pérez et al. [18] employed an explainable multi-label classification on Spanish legal judgments, achieving an F1 score of 82%. The authors showed that SHAP increased user trust for complex legal predictions. del Aguila Escobar et al. [19] proposed the OBOE (explanatiOns Based On concEpts) model, an interpretable machine learning framework for generating OBOE, grounded in the typicality interpretation of logic-based algorithms and the preference for explanation over precision–recall reporting. In the medical field, Veeranki et al. [20] explored multi-label text classification based on large-scale clinical real-world datasets, highlighting the effectiveness of machine learning models in healthcare applications. In a related medical context, Saleh and Yousif introduced a confidence-weighted rule-based framework for brain lesion classification using multimodal MRI and MRS data [21]. Additional contributions dealt with comparative and adversarial aspects. Zahrani demonstrated that XAI techniques can maintain high-accuracy classification results, even above 92% (spam detection), while interpretability is achieved through layers around them [22]. Cesarini et al. [23] also compared LIME and SHAP between datasets and proposed a post-hoc selection method for maximizing the credibility/believability as well as interpretability; however, this work did not report classification accuracy.

Conversely, Kozik et al. [24] demonstrated SHAP's limitations by failing in 86% of adversarial misclassification cases, casting doubt on the sole reliance on post-hoc explanations. Schneider et al. [25] continued this line of research by considering techniques for detecting and removing "fake" explanations, but without accompanying classification evaluations. HuLLMI planned to experiment with dataset sample sizes ranging from 10,000 to 100,000 and selected classical ML models (Naïve Bayes, MLP, Random Forest, and XGBoost) with T5 (transformer) embeddings. Transformer models are a promising avenue, as evidenced by non-transformer approaches (such as MLP, which achieves 88% accuracy), but LIME remains interpretable [26]. Consistent with these findings, a recent investigation in Scientific Reports compared logistic regression trained on large-scale injury narratives to ChatGPT-3.5 predictions, achieving a recall of 84%. We combined LIME with eye-tracking in the work to reveal that humans do agree and disagree on classification [20]. Most recently, continuing along this line, in 2025, Najjar et al. [27] proposed XAI-enhanced AI-based text detection in educational scenarios, using XGBoost with LIME and SHAP, achieving ~83% accuracy while also discovering machine-generated text-specific linguistic patterns. Wu et al. [28] provided a comprehensive review of LLM-based text detection, suggesting hybrid frameworks that reconcile the trade-off between performance and interpretability, but did not present new experimental results. Abolghasemi et al. [29] developed inductive learning systems for comparison with human and LLM-based decisions, finding 87% agreement between LLM outputs and human rules. Although they did not

directly utilize any XAI tools, their emphasis on interpretability in the domain of performance overlap aligns with the goals of the present work. Recent works also report on text representation strategies and model selection in AI text detection. Najjar et al. [27] obtained 83% accuracy on the CyberHumanAI training data by combining the TF-IDF and Bag-of-Words features with XGBoost and Random Forest. With LIME, the dominant difference was lexical between human and AI-produced documents [9, 18, 23]. In this paper, we aim to address this gap identified in prior studies, where a principled tradeoff between the performance and transparency of AI-generated text detection (especially multi-class).

3. METHODOLOGY

In the context of AI-based plagiarism detection, data preprocessing and representation are crucial for enhancing the effectiveness and accuracy of subsequent classification tasks. By refining the input data and selecting the most informative features, these steps help simplify the model's learning process and enhance its predictive performance. This study presents a structured framework for multi-class text classification and the detection of AI-generated texts—particularly those produced by LLMs—by distinguishing them from human-written content. As illustrated in Figure 1, the suggested framework consists of seven core stages: data collection, preprocessing, feature extraction, Data splitting, model training using XGBoost, evaluation, and interpretability through XAI techniques. Each of these components is discussed in detail in the following sections:

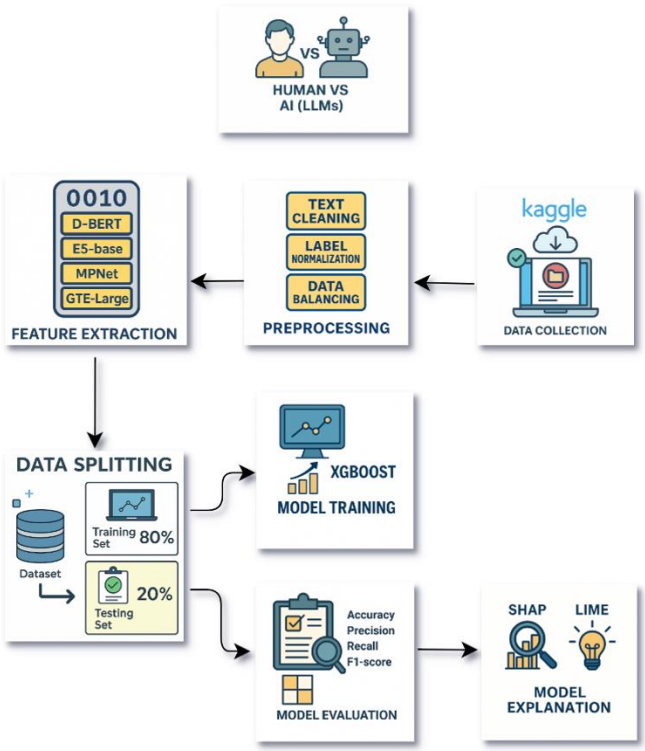


Figure 1. Workflow of XAI-based text classification: Human vs. Large Language Models (LLMs)

3.1 Dataset collection and description

The dataset employed in this work is sourced from the Kaggle database "Human vs. LLM Text Corpus" [30]. A raw dataset contained 788,922 source-labeled text samples in CSV file format with two underlying columns:

- **Text:** The textual content ranges from various domains such as technology, narrative, conversation, news, and official.
- **Generated:** A categorical label showing the generation source. The original dataset consisted of 62 unique label categories that corresponded to 61 sources of AI-generated text (including different versions and variants of LLMs, e.g., GPT-3, GPT-4, LLaMA-7B, LLaMA-13B, Claude-v1, Claude-v2, etc.) and another human-written texts category. Such fine-grained labeling is ideal for multi-class classification; however, class imbalance and a limited computational budget limited the number of labels that could be aggregated.

3.2 Pre-processing of data and sampling technique

The dataset was preprocessed across multiple stages to produce a balanced and computationally manageable subset for multi-class classification.

Stage 1- Cleaning and Normalisation

Text was lowercased, special characters/whitespace were removed, texts shorter than five words or samples that were too long or irrelevant were cleaned, the tag field was used to exclude unclassifiable and unencoded entries, and minimal normalization was applied to preserve the linguistic nuances relevant to AI text detection.

Stage 2- Class Consolidation

removed unknown or rare classes (less than 100 samples) trying to prevents overfitting (to Minority Classes), unified the classes, and reduced their number from 61 to 18 final AI-Generated classes, i.e., merging its from the same model family (e.g., GPT-3.5, GPT-4 → GPT), under Classification for Multi-class, this step was critical due to the potential overlap of text samples across multiple AI generators.

Stage 3- Random Oversampling

The dataset was then balanced using random oversampling to ensure balanced representation and prevent bias toward the more numerous types, resulting in a significantly larger total file size. This size increases enabled normalization of class distributions and balanced classification.

Stage 4- Stratified Sampling

The data was sampled stratifying by class, allowing us to work efficiently within the available computing resources. Figure 2 illustrates the reprocessing workflow.

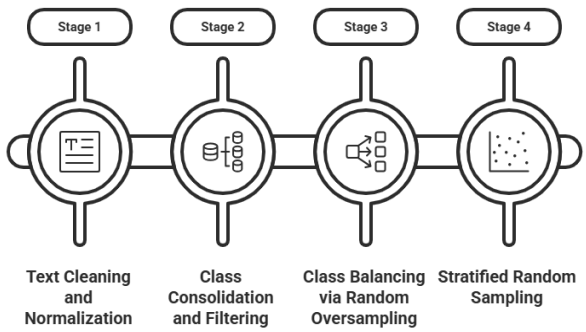


Figure 2. Pre-processing pipeline for dataset preparation

3.3 Embedding techniques

Text embedding is the process of converting unstructured text into fixed-length numerical vectors that capture not only

semantic meaning but also syntactic patterns and contextual relationships among words or sentences. Such representations enable machine learning models to comprehend and compare textual data in a mathematical space, facilitating tasks such as classification, clustering, retrieval, and analysis of semantic similarity. More modern embedding techniques, typically derived using neural networks, capture contextual knowledge; therefore, they provide better or more accurate language modeling than older bag-of-words or TF–IDF frameworks. Some of the newest embedding techniques included:

3.3.1 Distiled BERT

D-BERT is a smaller version of the original Bidirectional Encoder Representations from Transformers (BERT) [31], created via knowledge distillation, that maintains approximately the same language understanding ability as BERT with fewer parameters and lower computational cost. It achieves an efficiency-performance trade-off by leveraging predictions from a larger, pre-trained BERT model, which is particularly useful in resource-constrained environments or for large-scale applications that require real-time inference speed. It is a contextual embedding model, meaning that each word's representation depends on its surrounding words. This enables BERT to capture dependencies and similarities in meaning and structure between tokens in text. It has been demonstrated that these features are helpful for sentiment classification tasks, where not only accuracy but also inference time are crucial. In addition, D-BERT has also been modified for learning speech representations, with minimal performance drop in high-dimensional audio-derived text transcriptions, due to its compact size [32-34].

3.3.2 E5-base

A transformer-based embedding model fine-tuned for semantic recovery tasks. It can encode queries and passages into a shared vector space optimized for general-purpose text representation [35, 36]. The primary objective is to learn sentence representations that maximize semantic similarity for semantically related sentence pairs and minimize it for unrelated sentence pairs. A standard method for achieving this is contrastive learning, where the vector representations of two similar concepts (in representation space) are brought closer together by increasing their cosine similarity. Therefore, they can be abstractly represented by this formulation shown in Eq. (1):

$$sim(q,p) = \frac{q.p}{\|q\| \, \|p\|} \tag{1}$$

where, q and p are the embedding vectors of the query and passage, respectively, the positive pairs (relevant query-passage pairs) are brought closer together in the vector space, and negative pairs are kept far apart. It uses the transformer encoder architecture, and contextualized token embeddings are reduced to a fixed-sized sentence vector. In most cases, the aggregation is performed using mean pooling over the final hidden states. More recent developments have extended it to process long-context documents using self-extendable mechanisms, where long inputs are split into overlapping chunks whose representations are combined into a unified embedding.

3.3.3 MPNet

Associated the advantages of Masked Language Modeling (MLM), as seen in BERT, with Permuted Language Modeling (PLM), inspired by XLNet. This twin training technique can help MPNet to create high-quality semantic embeddings that better capture contextual dependence within text order. It has demonstrated strong performance in various natural language understanding tasks, thanks to its active encoding of both local and global semantic features [37, 38]. The intuition is quite simple: MLM replaces a subset of tokens in a sequence with a special [MASK] token and trains the model to predict these masked tokens based on the surrounding context. In contrast, PLM predicts tokens in a random permutation order to provide bi-directional dependencies without explicit masking. To build on their strengths, it employs a masking strategy in which tokens are first permuted and then fed into a mask generation mechanism that preserves relative positions while providing context continuity during simultaneous masking.

Formally, let X (x_1,x_2,..,x_n) be a sequence of tokens, π be a permutation over token indices, and $M \subset \pi$ be the set of masked positions. It then optimizes the following training objective, which can be abstractly as shown in Eq. (2):

$$\mathcal{L} = -\sum_{i \in M} log \; P_\theta(x_i \mid X_{\setminus M}, \pi) \tag{2}$$

where, $X_{\setminus M}$ denotes the sequence with masked tokens removed, and P_θ is a probability distribution over the vocabulary parameterized by the model. This permutation π ensures that deviations between masked and unmasked tokens are balanced, thereby overcoming the mask-independence issue of BERT's MLM. It encodes sequences using a transformer encoder, generating contextual embeddings that capture both local (short-range) and global (long-range) semantic features. Then, a number of these embeddings are aggregated into a fixed-sized vector using mean pooling for downstream tasks such as Semantic Textual Similarity, text classification, or retrieval [39].

3.3.4 General Text Embeddings – Large

GTE-Large is a Roberto NLP model built upon transformers and meant for tasks more complex than simple question-answering, such as text classification. GTE belongs to the most recent embedding family of models, reducing complexity and dimensionality by leveraging large-scale transformer encoders to capture contextualized word dependencies [32], similar to how Self-Attention mechanisms operate on unstructured context. The Large variant uses more layers, hidden dimensions, and attention heads than base models to model complex linguistic patterns and semantic nuances [40, 41]. Essentially, it works by feeding input sequences through the transformer encoder, which produces a sequence of hidden states. A fixed-length sentence embedding can be obtained by aggregating these token-level hidden states with mean pooling over the final layer, which can be abstractly represented by the formulation in Eq. (3):

$$s = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} h_i \tag{3}$$

where, h_i is the hidden state representation of the i^{th} token, and n is the sequence length. It creates a sentence vector s, which is a point in the high-dimensional continuous space, and similar sentences are closer to one another.

3.4 Data splitting

For assessment of model performance, the training and test data were partitioned using a fixed-random split (Hold-out). Appropriate data splitting is an important aspect of machine learning, as the proportion of training versus testing data can have a considerable impact on model performance and reproducibility. A (Train: 118,267; Test: 29,567) split ratio was employed, as it follows common practice and experimental design recommendations to balance learning capability and fair testing [42, 43]. Additionally, measures were taken to ensure that the distribution of classes across subsets remained balanced, thereby preventing data imbalance. Studies have also confirmed the importance of optimal splitting techniques in reducing sample bias and improving the reproducibility of experimental results [44].

3.5 Model training (XGBoost classifier)

We have chosen the XGBoost (Extreme Gradient Boosting) model for classification because of its excellent skill, scalability, and robust predictive accuracy. It is efficient for processing structured data and supports parallel computation, regularization better tree boosting [45]. It has been effectively used in many fields, including environmental modeling [46] and civil engineering [47], demonstrating its versatility. Additionally, merging XGBoost with SHAP-based explanation methods has been shown to provide valuable insights into feature importance and model decision-making, making it a dependable choice for XAI systems. It is widely used for its accuracy. It builds boosted trees iteratively, minimizing an objective function that combines a loss term and a regularization term, thereby improving generalization and reducing overfitting. Its objective function combines loss minimization and tree regularization, shown in Eq. (4):

$$\mathcal{L}(\phi) = \sum l\left(y_i, \hat{y}_i^{(t)}\right) + \sum \Omega(f_k) \tag{4}$$

where,

- l: loss function (e.g., log-loss)
- y_i: true label, $\hat{y}_i$: predicted output
- $\Omega(f)$: regularization term for tree complexity

In practice, XGBoost approximates this objective using a second-order Taylor expansion of the loss function, making it computationally efficient for finding gains from tree splits. In addition, it uses L1 and L2 regularization, making this model less prone to overfitting than the traditional boosting model. XGBoost natively handles missing values and parallelizes training, improving its scalability for large datasets.

3.6 Model evaluation metrics

To assess classification model performance, a set of standard evaluation metrics is used: accuracy, precision, recall, and F1 score. It estimates a model's predictive capability by calculating the rates of correct and incorrect predictions [48]. The nature of this dataset, with multiple classes and potential class imbalance, also revealed that total averages and weighted F1 scores were needed to achieve balanced performance [49-51]. The evaluation metrics used in this study are defined in Eqs. (5)-(8):

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{5}$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{6}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{7}$$

$$\text{F1} - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

3.7 Explainability techniques (SHAP, LIME)

To improve model interpretability and interpret the predictions of classification models, two popular XAI methods [52] are employed: SHAP (SHapley Additive exPlanations) [12, 53] and LIME (Local Interpretable Model-Agnostic Explanations) [11]. Such techniques reveal the contributions of input features or tokens to model decisions and help identify decision-relevant features, thereby increasing trust in the system. SHAP and LIME have been widely used across many domains and tested on various tasks, demonstrating their reliability and stability in high-dimensional classification. SHAP supports both global and local interpretability. Each of the input variables is given with an additive feature importance score, which allows for visualizing and explaining decisions, as in Eq. (9):

$$f(x) = \varphi_0 + \sum \varphi_i \tag{9}$$

where,

$f(x)$: model prediction,
φ_0: base value (expected output)
φ_i: SHAP value for feature i

The simplified formulation of SHAP values highlights how each feature influences a prediction by estimating its marginal contribution compared to all possible feature subsets, as shown in Eq. (10):

$$\varphi_i = \sum [f(S \cup \{i\}) - f(S)] \, . \, W(S) \tag{10}$$

where,

S: subset of input features,
$f(S)$: model output on feature subset,
$w(S)$: weighting function

While LIME describes the model locally using interpretable replacement models to explain individual predictions, as shown in Eq. (11):

$$g^* = arg \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{11}$$

where,

g: interpretable model,
G: set of all interpretable models
π_x: locality kernel around instance x,
$\Omega(g)$: model complexity penalty

In practice, LIME samples perturbations around the instance of interest, asks the Blackbox model for predictions, and then fits a simple interpretable model (for instance, a sparse linear regressor) weighted by the locality kernel π_x. This ensures that the explanation focuses on the local decision boundary rather than the global model. Nevertheless, LIME explanations can exhibit variance across runs when random perturbations are used, and they may also be sensitive to both kernel width and sampling strategy, limiting their stability in some instances.

4. RESULTS AND DISCUSSION

This section reports the evaluation of the proposed framework for classifying human-written and LLM-generated text. The results combine quantitative metrics from four Transformer-based embeddings with an XGBoost classifier, along with qualitative explanations derived from XAI techniques. We evaluate the performance of various sentence embedding methods for the XGBoost classifier and then compare accuracy, precision, recall, and F1-score. Explainability techniques (SHAP, LIME) are then used to interpret model predictions and further support the transparency of the classification process by revealing how individual text features contribute to the outcome. Finally, the language patterns for each label are investigated using visualization methods, including SHAP force plots, LIME local explanations, and word clouds.

4.1 Performance and comparative analysis

This section compares four Transformer-based embedding models integrated with the XGBoost classifier. All models utilize different sentence embedding techniques, i.e., DistilBERT, MPNet, E5-base, and GTE-Large. The evaluation is based on standard classification metrics—accuracy, precision, recall, and F1-score—as well as interpretability feedback from the explainability tools SHAP and LIME. The interpretability and classification results of each model are tabulated in Table 1. Among the evaluated methods, E-5 base performed best, with an accuracy of 89%, followed closely by GTE-Large (88%). DistilBERT and MPNet both achieved 87%. Interpretability-wise, SHAP analysis showed that the embedding-based model generated semantically consistent and meaningful attribute values, which are easier to comprehend in terms of how they led to the prediction. Additionally, LIME explanations identified keywords and phrases that most influenced classification decisions, thereby increasing transparency and credibility. These results suggest that the E5-base and GTE-Large models strike the best balance between representation quality and interpretability, making them most suitable for multi-class classification of AI-generated versus human-written text.

Table 1. Performance and explainability summary of transformer-based embedding with XGBoost

Model	Algorithm	Accuracy	Precision	Recall	F1-Score	SHAP Insights	LIME Insights
D_BERT	XGBOOST	0.87	0.87	0.87	0.87	Several embedding dimensions in D_BERT had strong SHAP values	Key text tokens influenced the prediction
E5_base	XGBOOST	0.89	0.88	0.89	0.88	Key E5_base dimensions contributed significantly to predictions	Semantic phrases contributed to the label decision
MpNet	XGBOOST	0.87	0.87	0.87	0.87	MPNet had localized importance with a few dominant dimensions	Keywords highlighted model reasoning
GTE_Large	XGBOOST	0.88	0.88	0.88	0.88	Balanced SHAP values across many dimensions in GTE_Large	Typical label-aligned phrases explained predictions

4.1.1 Class-wise performance analysis

Table 2. Class-wise F1-score comparison across embedding models

Class	D-BERT F1	MPNet F1	E5-Base F1	GTE-Large F1
Bloom	0.95	0.98	0.98	0.99
Claude	0.98	0.98	0.98	0.98
Falcon	0.91	0.91	0.91	0.91
Flan	0.71	0.69	0.72	0.69
GLM	0.97	0.97	0.98	0.98
GPT	0.72	0.70	0.74	0.74
Human	0.56	0.61	0.62	0.59
LLaMA	0.79	0.79	0.83	0.79
LZLV	0.98	0.98	0.98	0.98
Mistral	0.96	0.96	0.96	0.96
Mixtral	0.94	0.94	0.94	0.94
Neural	1.00	1.00	1.00	1.00
Nous	0.99	0.99	0.99	0.99
OPT	0.64	0.62	0.67	0.64
PaLM	0.99	0.99	0.99	0.99
Psyfighter	0.99	0.99	0.99	0.99
T0-11B	0.92	0.92	0.93	0.93
Text	0.68	0.67	0.71	0.69

Apart from overall performance, we also calculated fine-grained class metrics for all 18 entities (Human + 17 LLMs). The models have also shown significant distinctions among the LLM families (F1-Score F1 = [0.59 - 1.00]), demonstrating that the system can indeed identify which text belongs to a particular generator, rather than only whether it was generated by Human vs AI. The classification performance across all 18 LLM categories and four embedding models is summarized in Table 2, indicating distinctive separability across different LLM families.

The Class-Wise scores indicate that many LLMs, such as Neural, PaLM, Claude, and Psyfighter (F1 ≈ 0.99 – 1.00), achieved perfect separability, meaning they are very distinctive from each other; meanwhile, Human, GPT, and FlanOPT. Moreover, texts achieve lower F1 scores because they tend to be more human-like and conversational. This indicates that the model is performing multi-class authorship attribution, not just binary Human vs. AI classification.

4.2 Baseline comparison with a fine-tuned DistilRoBERTa model

To assess the suggested framework against a transformer-based state-of-the-art baseline, a DistilRoBERTa model was fine-tuned on the same dataset. The baseline achieved 88% accuracy, an F1-score of 87.9%, precision of 88%, recall of 88%, and an inference time of 34 ms per sample. In comparison, the proposed E5-base + XGBoost model achieves a slightly higher accuracy (89%) while being significantly faster (8 ms per sample) and offering full explainability through SHAP and LIME. This accuracy-speed-explainability balance demonstrates the practicality and interpretability

advantage of our approach over transformer-only detectors.

A summary is provided in Table 3, while detailed evaluation metrics of the baseline model are included in Appendix D.

Table 3. Performance comparison between the proposed model and DistilRoBERTa baseline

Model	Accuracy	Precision	Recall	F1-Score	Inference Time
E5-base + XGBoost	**0.89**	0.88	0.89	0.88	**8 ms**
DistilRoBERTa (Fine-tuned)	0.88	0.88	0.88	0.879	34 S

Table 4. Comparative of recent studies on artificial intelligence generated text detection using XAI

Study	Dataset	Models/Algorithms	XAI Techniques	Accuracy	Notes
Our study	Human vs. LLM Text Corpus (Kaggle) 147,834 Balanced samples (18 classes: Human + 17 Large Language Models (LLMs))	DistilBERT, MPNet, E5-base, GTE-large, XGBoost	SHAP, LIME	E5-base: 89%	Multi-class classification distinguishing between human-written and 17 distinct LLM-generated text types with explainable AI
Najjar et al. [27]	CyberHumanAI (500 humans, 500 ChatGPT texts)	TF-IDF, Bag-of-Words XGBoost, Random Forest	LIME	83%	Used LIME to identify distinguishing features between human and AI-created texts, noting differences in word usage patterns
HuLLMI (2024)	Multiple datasets, including curated corpora and real-world samples 10,000–100,000 range, depending on the sub-dataset.	TF-IDF, Bag-of-Words Naïve Bayes, MLP, Random Forest, XGBoost For T5 Modern Transformer Embedding	LIME	XGBoost: 72%, MLP: 88%, T5: 88%	Proved that traditional ML models perform comparably to modern NLP detectors in human vs. AI text detection, with LIME providing explainability.
Scientific Reports (2024)	204 injury narratives	TF-IDF Logistic Regression (trained on 120,000 samples), ChatGPT-3.5	LIME, Eye-tracking	ML model Recall: 84%	Compared human and artificial intelligence model performance and explainability in text classification tasks, focusing on fields of agreement and difference
Cesarini et al. [23]	Different textual datasets Training corpus: 120,000 labeled samples	traditional (TF-IDF) and transformer embeddings (BERT-like), Evaluation of post-hoc XAI methods	SHAP, LIME, and others	undefined	Presented an overall evaluation of different XAI methods in text classification, focusing on explainability and user confidence
de Arriba-Pérez et al. [18]	Spanish legal judgments several thousand to tens of thousands, depending on each benchmark.	Spanish BERT model Multi-label classification models	visible and descriptive explanations	85% micro-accuracy	Progressed a method collected with ML and natural explanations to classify legal judgments effectively.

4.3 Explainability techniques (XAI integration)

4.3.1 SHAP visualizations and interpretations

Figure A1 presents the SHAP visual outputs in Appendix A in a clear and orderly manner for all four models. We can see which dimensions consistently affect the prediction. The printed colors of the stacked bars illustrate which classes depend the most on each dimension, providing insight into both the importance of each dimension and specific-class dependencies. The broad distribution of importance across many classes indicates models capture a wide range of semantic signals, rather than favoring one dominant class.

Overall, SHAP summaries suggest potentially complementary behaviours among the four embedding models. In contrast, D-BERT shows less localised feature importance, indicating that shared semantic dimensions are more widely distributed across classes. MPNet and E5-base are more focused on class-specific dimensions, whereas attention is more dispersed in GTE-Large, distributing importance more generally, which leads to robustness against well-discriminated class layers. Complete per-model SHAP interpretations are available in Appendix A.

4.3.2 LIME explanations

Figure A2 presents the LIME visual outputs in Appendix B for all four models. Offered token-level interpretability, focus on the main words and phrases that contribute to the classification decision. These insights help us understand how different models support linguistic patterns across human and AI-generated text categories, using various LLMs. Transformer-based models exhibited more consistent patterns, where semantically meaningful tokens (e.g., verbs, technical terms) were associated with higher prediction weights. In particular, GTE-Large provided the most straightforward explanation by clearly highlighting the effective tokens for each output label, thereby supporting its superior

interpretability. These LIME-based explanations complement the SHAP global explanations by providing precise, local, and reasonable explanations, thereby fostering deeper trust in the model's actions.

At the token level, LIME confirms that all four models attend to semantically meaningful words and phrases, but with different degrees of focus. GTE-Large produces the most sharply localised token attributions, and MPNet offers a balanced mixture of contextual and keyword-based evidence. At the same time, D-BERT spreads importance more diffusely, and E5-base also assigns weight to function words. These patterns complement the global SHAP findings and are reported in total detail in Appendix B.

4.3.3 Comparison of SHAP and LIME interpretations

The combination of both SHAP and LIME offered a dual perspective on model behavior. SHAP provided a global view, identifying the most impactful embedding dimensions that drove the model's predictions, while LIME offered local, human-interpretable, and visually appealing explanations, presenting words and phrases that affected individual predictions. A comparison of the two interpretability methods revealed strong alignment: tokens with high SHAP contributions from specific dimensions are often highlighted by LIME as meaningful. Such alignment between numerical and textual explanations also helps establish confidence in the model's overall reasoning process, which is crucial for trust, especially in high-stakes applications. Additional qualitative examples illustrating this alignment are included in Appendices A and B.

4.4 Confusion matrix interpretation

To further assess class-level behaviour, confusion matrices were generated for the four embedding-based classifiers (DistilBERT, MPNet, E5-base, GTE-Large). Figure A3 shows the visualized results. Complete confusion matrices are available in Appendix C for transparency and reproducibility. For all models, most LLM classes strongly dominate the diagonals of this matrix, indicating high discriminability between generators such as Neural, PaLM, Claude, Bloom, Mistral, and Nous. The confusing patterns are essentially between Humans, GPT, Flan, OPT, and Text –all of which can be positioned close to human-like instruction-following and conversational styles. This consistency indicates that the classification challenge is not an algorithmic weakness, but genuine overlap in linguistic style between these specific label groups.

4.5 Benchmarking against prior studies

Table 4 situates this work within the broader context of a recent survey of studies on explainable machine learning methods for detecting AI-generated texts. Traditional machine learning models, combined with interpretability tools, have been successfully used in previous work. Here, the novelty lies in the inclusion of multiple state-of-the-art sentence-embedding transformers used in conjunction with an XGBoost classifier. Comparing this work with previous studies, which mainly employ binary classification or a single embedding approach, this work treats multi-class classification as a more challenging task in terms of real AI-generated diversity. Moreover, by combining global (SHAP) and local (LIME) explainability methods, a deeper level of transparency is achieved, providing more insights into model behavior across various text sources. differentiating it from previous attempts in that it combines accuracy and interpretability.

5. CONCLUSION

In this paper, we introduce and test a system for interpreting machine learning approaches to the classification of human- and machine-generated texts using a variety of sentence embeddings and explainable AI algorithms. We show that transformer-based embeddings, specifically E5-base and GTE-Large, achieve superior classification performance, thereby further improving model interpretability. By combining SHAP for global and LIME for local token-level interpretations, we gained meaningful insights into how specific textual properties and embedding dimensions influence the model's decisions. These complementary interpretation techniques not only reinforce trust in the model's decision-making but also demonstrate the subtle linguistic signals that distinguish between human- and machine-generated language. In summary, this work highlights the importance of integrating high-performing embeddings with transparent explanation frameworks, leading to more interpretable, accountable, and generalizable AI text detection systems.

REFERENCES

[1] Yousif, S.A., Samawi, V.W., Elkabani, I. (2017). Arabic text classification: The effect of the AWN relations weighting scheme. In Proceedings of the World Congress on Engineering 2017 Vol II, London, U.K., pp. 1-5. https://www.iaeng.org/publication/WCE2017/WCE2017_pp594-598.pdf.

[2] Jehad, R., Yousif, S.A. (2020). Fake news classification using random forest and decision tree (J48). Al-Nahrain Journal of Science, 23(4): 49-55. https://doi.org/10.22401/ANJS.23.4.09

[3] Sutriawan, Sasoko, W.H., Alamin, Z., Ritzkal. (2025). Benchmarking text embedding models for multi-dataset semantic textual similarity: A machine learning-based evaluation framework. Acadlore Transactions on AI and Machine Learning, 4(2): 82-96. https://doi.org/10.56578/ataiml040202

[4] Ahmed, A.S., Haddad, A.A.A., Hameed, R.S., Taha, M.S. (2025). An accurate model for text document classification using machine learning techniques. Ingenierie des Systèmes d'Information, 30(4): 913-921. https://doi.org/10.18280/isi.300408

[5] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 3: 1137-1155. https://dl.acm.org/doi/10.5555/944919.944966.

[6] Yang, Z.W., Feng, Z.J., Huo, R.X., Lin, H.R., Zheng, H.H., Nie, R.C., Chen, H.R. (2025). The imitation game revisited: A comprehensive survey on recent advances in AI-generated text detection. Expert Systems with Applications, 272: 126694. https://doi.org/10.1016/j.eswa.2025.126694

[7] Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., et al. (2023). Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy

artificial intelligence. Information Fusion, 99: 101805. https://doi.org/10.1016/j.inffus.2023.101805

[8] Martins, T., de Almeida, A.M., Cardoso, E., Nunes, L. (2024). Explainable artificial intelligence (XAI): A systematic literature review on taxonomies and applications in finance. IEEE Access, 12: 618-629. https://doi.org/10.1109/ACCESS.2023.3347028

[9] Najjar, A., Ashqar, H.I., Darwish, O.A., Hammad, E. (2025). Leveraging explainable AI for LLM text attribution: Differentiating human-written and multiple LLMs-generated text. arXiv preprint arXiv:2501.03212. https://doi.org/10.48550/arXiv.2501.03212

[10] Salih, A., Raisi-Estabragh, Z., Galazzo, I.B., Radeva, P., Petersen, S.E., Menegaz, G., Lekadir, K. (2023). A perspective on explainable artificial intelligence methods: SHAP and LIME. arXiv preprint arXiv:2305.02012. https://doi.org/10.48550/arXiv.2305.02012

[11] Ribeiro, M.T., Singh, S., Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, pp. 1135-1144. https://doi.org/10.1145/2939672.2939778

[12] Lundberg, S.M., Lee, S.I. (2017). A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, pp. 4768-4777. https://dl.acm.org/doi/10.5555/3295222.3295230.

[13] Khosravi, H., Buckingham Shum, S., Chen, G., Conati, C., et al. (2022). Explainable artificial intelligence in education. Computers and Education: Artificial Intelligence, 3: 100074. https://doi.org/10.1016/j.caeai.2022.100074

[14] Weng, L.X., Zhu, M.F., Wong, K.K., Liu, S., Sun, J.S., Zhu, H., Han, D.M., Chen, W. (2023). Towards an understanding and explanation for mixed-initiative artificial scientific text detection. arXiv preprint arXiv:2304.05011. https://doi.org/10.48550/arXiv.2304.05011

[15] Shah, A., Ranka, P., Dedhia, U., Prasad, S., Muni, S., Bhowmick, K. (2023). Detecting and unmasking AI-generated texts through explainable artificial intelligence using stylistic features. International Journal of Advanced Computer Science and Applications (IJACSA), 14(10): 1043-1053. https://doi.org/10.14569/IJACSA.2023.01410110

[16] Hajialigol, D., Liu, H.W., Wang, X. (2023). XAI-CLASS: Explanation-enhanced text classification with extremely weak supervision. arXiv preprint arXiv:2311.00189. https://doi.org/10.48550/arXiv.2311.00189

[17] Zahoor, K., Bawany, N.Z., Qamar, T. (2024). Evaluating text classification with explainable artificial intelligence. IAES International Journal of Artificial Intelligence, 13(1): 278-286. https://doi.org/10.11591/ijai.v13.i1.pp278-286

[18] de Arriba-Pérez, F., García-Méndez, S., González-Castaño, F.J., González-González, J. (2024). Explainable machine learning multi-label classification of Spanish legal judgements. arXiv preprint arXiv:2405.17610. https://doi.org/10.48550/arXiv.2405.17610

[19] del Águila Escobar, R.A., Suárez Figueroa, M.C., Fernández López, M. (2024). OBOE: An explainable text classification framework. International Journal of Interactive Multimedia and Artificial Intelligence, 8(6): 24-37. https://doi.org/10.9781/ijimai.2022.11.001

[20] Veeranki, S.P.K., Abdulnazar, A., Kramer, D., Kreuzthaler, M., Lumenta, D.B. (2024). Multi-label text classification via secondary use of large clinical real-world data sets. Scientific Reports, 14: 26972. https://doi.org/10.1038/s41598-024-76424-8

[21] Saleh, S., Yousif, S.A. (2025). A confidence-weighted rule-based framework for multimodal brain lesion classification using MRI and MRS. Ingenierie des Systèmes d'Information, 30(6): 1579-1587. https://doi.org/10.18280/isi.300616

[22] Alzahrani, A. (2024). Explainable AI-based framework for efficient detection of spam from text using an enhanced ensemble technique. Engineering, Technology & Applied Science Research, 14(4): 15596-15601. https://doi.org/10.48084/etasr.7901

[23] Cesarini, M., Malandri, L., Pallucchini, F., Seveso, A., Xing, F. (2024). Explainable AI for text classification: Lessons from a comprehensive evaluation of post hoc methods. Cognitive Computation, 16: 3077-3095. https://doi.org/10.1007/s12559-024-10325-w

[24] Kozik, R., Ficco, M., Pawlicka, A., Pawlicki, M., Palmieri, F., Choraś, M. (2024). When explainability turns into a threat - using XAI to fool a fake news detection method. Computers & Security, 137: 103599. https://doi.org/10.1016/j.cose.2023.103599

[25] Schneider, J., Meske, C., Vlachos, M. (2024). Deceptive XAI: Typology, creation, and detection. SN Computer Science, 5: 81. https://doi.org/10.1007/s42979-023-02401-z

[26] Joshi, P.D., Pocker, S., Dandekar, R.A., Dandekar, R., Panat, S. (2024). HULLMI: Human vs LLM identification with explainability. arXiv preprint arXiv:2409.04808. https://doi.org/10.48550/arXiv.2409.04808

[27] Najjar, A.A., Ashqar, H.I., Darwish, O.A., Hammad, E. (2025). Detecting AI-generated text in educational content: Leveraging machine learning and explainable AI for academic integrity. arXiv preprint arXiv:2501.03203. https://doi.org/10.48550/arXiv.2501.03203

[28] Wu, J.C., Yang, S., Zhan, R.Z., Yuan, Y.L., Chao, L.S., Wong, D.F. (2025). A survey on LLM-generated text detection: Necessity, methods, and future directions. Computational Linguistics, 51(1): 275-338. https://doi.org/10.1162/coli_a_00549

[29] Abolghasemi, M., Ganbold, O., Rotaru, K. (2025). Humans vs. large language models: Judgmental forecasting in an era of advanced AI. International Journal of Forecasting, 41(2): 631-648. https://doi.org/10.1016/j.ijforecast.2024.07.003

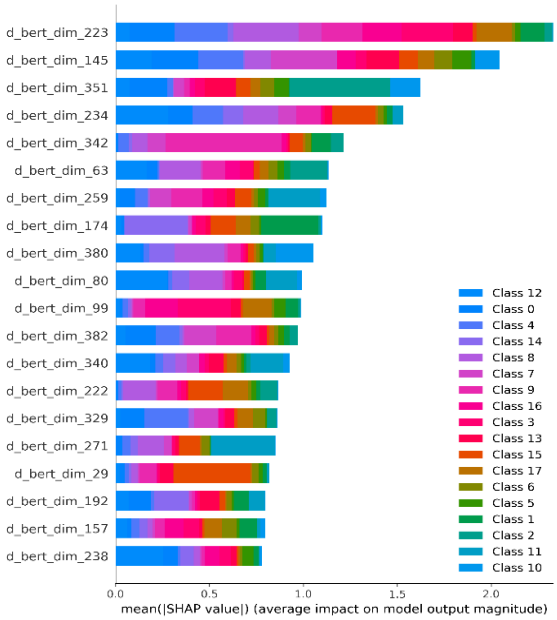
[30] Grinberg, Z. (2023). Human vs. LLM text corpus. https://www.kaggle.com/datasets/starblasters8/human-vs-llm-text-corpus.

[31] Sun, C., Huang, L.Y., Qiu, X.P. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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), Minneapolis, Minnesota, pp. 380-385.

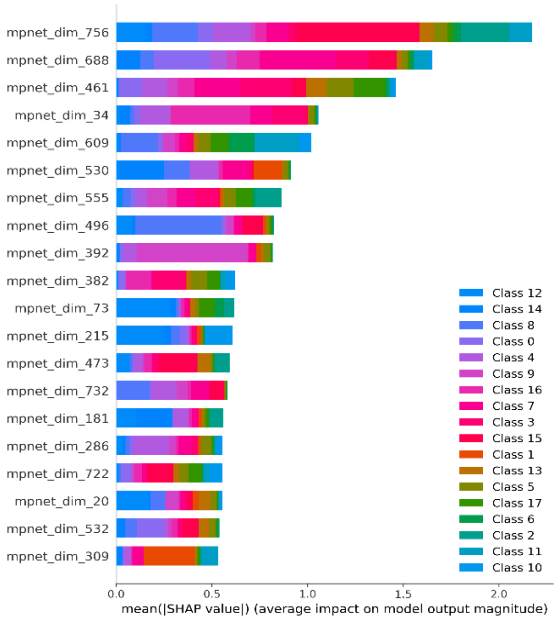
https://doi.org/10.18653/v1/N19-1035
[32] Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108
[33] Wang, H.Y., Kang, X., Ren, F.J. (2022). Emotion-Sentence-DistilBERT: A sentence-BERT-based distillation model for text emotion classification. In Artificial Intelligence and Robotics. ISAIR 2022. Communications in Computer and Information Science, pp. 313-322. https://doi.org/10.1007/978-981-19-7943-9_27
[34] Yu, F., Guo, J., Xi, W., Yang, Z., Jiang, R., Zhang, C. (2021). Audio DistilBERT: A distilled audio BERT for speech representation learning. In 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, pp. 1-8. https://doi.org/10.1109/IJCNN52387.2021.9533328
[35] Zhu, D.W., Wang, L., Yang, N., Song, Y.F., Wu, W.H., Wei, F.R., Li, S.J. (2024). LongEmbed: Extending embedding models for long context retrieval. arXiv preprint arXiv:2404.12096. https://doi.org/10.48550/arXiv.2404.12096
[36] Liapis, C.M., Kyritsis, K., Perikos, I., Paraskevas, M. (2024). Transformer-based embeddings for Greek language categorization. In 2024 IEEE/ACIS 24th International Conference on Computer and Information Science (ICIS), Shanghai, China, pp. 176-181. https://doi.org/10.1109/ICIS61260.2024.10778332
[37] Song, K.T., Tan, X., Qin, T., Lu, J.F., Liu, T.Y. (2020). MPNet: Masked and permuted pre-training for language understanding. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, pp. 16857-16867. https://dl.acm.org/doi/10.5555/3495724.3497138.
[38] Li, L.J., Miao, Y.L., Qureshi, A.H., Yip, M.C. (2021). MPC-MPNet: Model-predictive motion planning networks for fast, near-optimal planning under kinodynamic constraints. IEEE Robotics and Automation Letters, 6(3): 4496-4503. https://doi.org/10.1109/LRA.2021.3067847
[39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. In 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, pp. 6000-6010. https://dl.acm.org/doi/10.5555/3295222.3295349.
[40] Thenlper. (2023). GTE-Large: General text embeddings. https://huggingface.co/thenlper/gte-large, accessed on Dec. 7, 2025.
[41] Memduhoğlu, A., Fulman, N., Zipf, A. (2024). Enriching building function classification using large language model embeddings of OpenStreetMap tags. Earth Science Informatics, 17: 5403-5418. https://doi.org/10.1007/s12145-024-01463-8
[42] Rácz, A., Bajusz, D., Héberger, K. (2021). Effect of dataset size and train/test split ratios in QSAR/QSPR multiclass classification. Molecules, 26(4): 1111. https://doi.org/10.3390/molecules26041111
[43] Nguyen, Q.H., Ly, H.B., Ho, L.S., Al-Ansari, N., Le, H.V., Tran, V.Q., Prakash, I., Pham, B.T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. Mathematical Problems in Engineering, 2021: 4832864. https://doi.org/10.1155/2021/4832864
[44] Joseph, V.R., Vakayil, A. (2022). SPlit: An optimal method for data splitting. Technometrics, 64(2): 166-176. https://doi.org/10.1080/00401706.2021.1921037
[45] Chen, T.Q., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, pp. 785-794. https://doi.org/10.1145/2939672.2939785
[46] Zhang, J.Y., Ma, X.L., Zhang, J.L., Sun, D.L., Zhou, X.Z., Mi, C.L., Wen, H.J. (2023). Insights into geospatial heterogeneity of landslide susceptibility based on the SHAP-XGBoost model. Journal of Environmental Management, 332: 117357. https://doi.org/10.1016/j.jenvman.2023.117357
[47] Nguyen, N.H., Abellán-García, J., Lee, S., Garcia-Castano, E., Vo, T.P. (2022). Efficient estimating compressive strength of ultra-high performance concrete using XGBoost model. Journal of Building Engineering, 52: 104302. https://doi.org/10.1016/j.jobe.2022.104302
[48] Rainio, O., Teuho, J., Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. Scientific Reports, 14: 6086. https://doi.org/10.1038/s41598-024-50929-w
[49] Vujović, Ž.Đ. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications (IJACSA), 12(6): 599-606. https://doi.org/10.14569/IJACSA.2021.0120670
[50] Naidu, G., Zuva, T., Sibanda, E.M. (2023). A review of evaluation metrics in machine learning algorithms. In Artificial Intelligence Application in Networks and Systems. CSOC 2023. Lecture Notes in Networks and Systems, pp. 15-25. https://doi.org/10.1007/978-3-031-35314-7_2
[51] Matti, R.S., Yousif, S.A. (2023). AutoKeras for fake news identification in Arabic: Leveraging deep learning with an extensive dataset. Al-Nahrain Journal of Science, 26(3): 60-66. https://doi.org/10.22401/ANJS.26.3.09
[52] Vimbi, V., Shaffi, N., Mahmud, M. (2024). Interpreting artificial intelligence models: A systematic review on the application of LIME and SHAP in Alzheimer's disease detection. Brain Informatics, 11: 10. https://doi.org/10.1186/s40708-024-00222-1
[53] Roshan, K., Zafar, A. (2022). Using kernel SHAP XAI method to optimize the network anomaly detection model. In 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, pp. 74-80. https://doi.org/10.23919/INDIACom54597.2022.9763241

APPENDIX A. SHAP Global Explanations

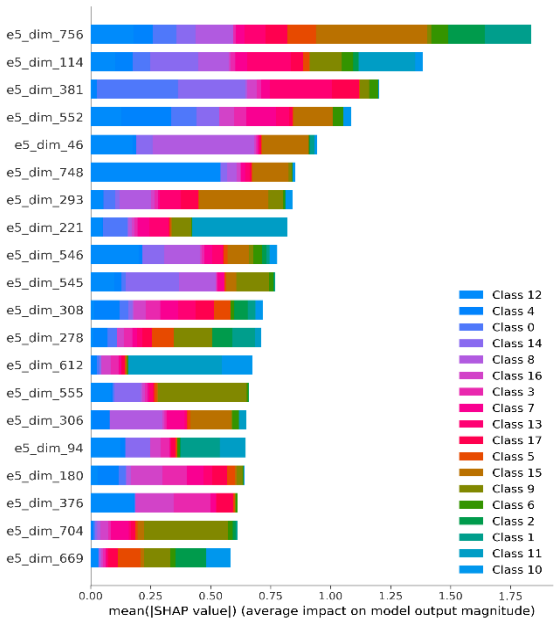
This appendix provides the full-resolution SHAP summary plots for all four embedding-based XGBoost models. These figures complement Figure A1 in the main text.



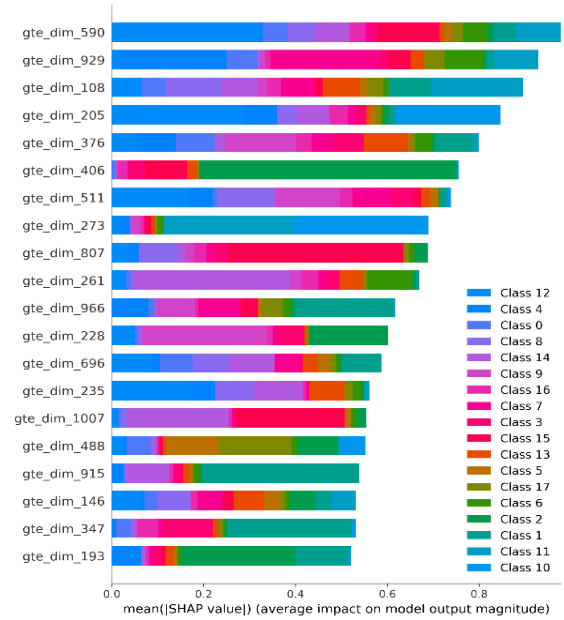
(a) D-BERT



(b) MPNet



(c) E5-base



(d) GTE-Large

Figure A1. SHAP visualizations

In more detail, the SHAP interpretations for each embedding model can be described as follows:

DistilBERT (Figure A1(a))

- The features with the highest degree of influence (e.g., dim_223, dim_145, dim_351) make strong contributions to numerous classes.
- SHAP bars from DistilBERT exhibit extensive color mixing, implying that its embeddings represent commonalities between classes, instead of dimensions that are highly class-specific.
- This means that it may generalize well, albeit with less discriminative power for fine-grained class separation, compared to other models.

MPNet (Figure A1(b))

- The most-occurring features (e.g., dim_756, dim_688, dim_461) present more focused contributions to certain classes (especially Classes 7, 13, 16).
- MPNet exhibits more class dependence than DistilBERT, indicating that it has a stronger capability to encode discriminative features.
- This is consistent with MPNet's design (masked + permuted training), which is more likely to encode local and global dependencies in a more balanced manner.

E5-base (Figure A1(c))

- E5-base has a small number of very dominant dimensions (dim_756, dim_114), where individual features have large effects on predictions for many classes.
- Unlike DistilBERT, the attention is more unevenly balanced — apparently, some dimensions serve more as semantic bottlenecks, bearing a greater predictive load.
- (which could be attributed to the task-specific fine-tuning of E5-base embeddings for semantic similarity, favoring compact, transparent representations but potentially relying too much on a few features).

GTE-Large (Figure A1(d))

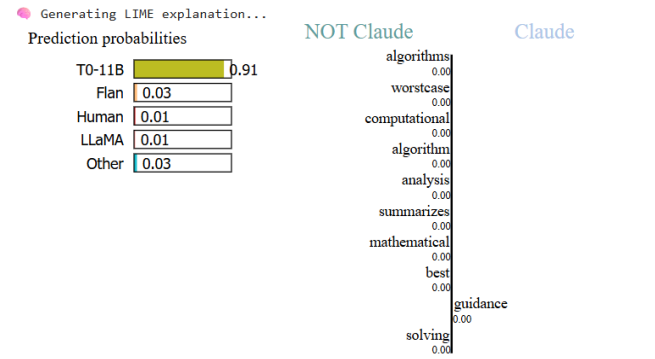
- GTE-Large has overall lower SHAP magnitudes (max

≈ 0.8) which suggests that it tends to distribute its predictions more uniformly over many dimensions.

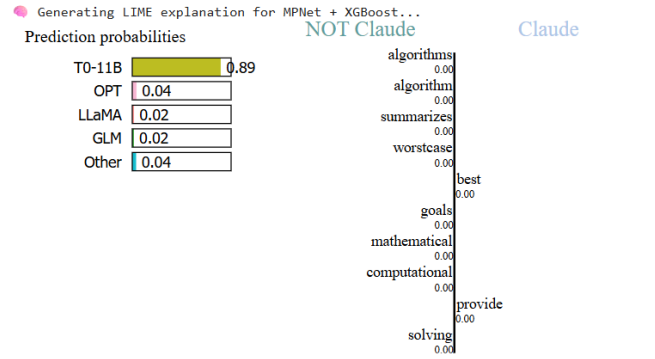
- This "flat" nature of the importance profile means the model is robust (redundant): it is not over reliant on any one dimension.
- GTE-Large further demonstrates more well-separated class-specific contributions, (e.g., Classes 12, 7, 9) and explanations at the class level are more interpretable.

Appendix B. LIME Token Attribution Results

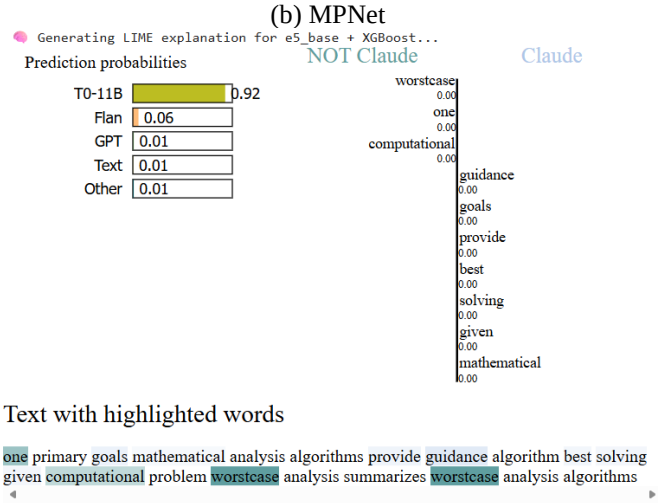
This section contains the complete LIME outputs for representative samples across all 18 classes.



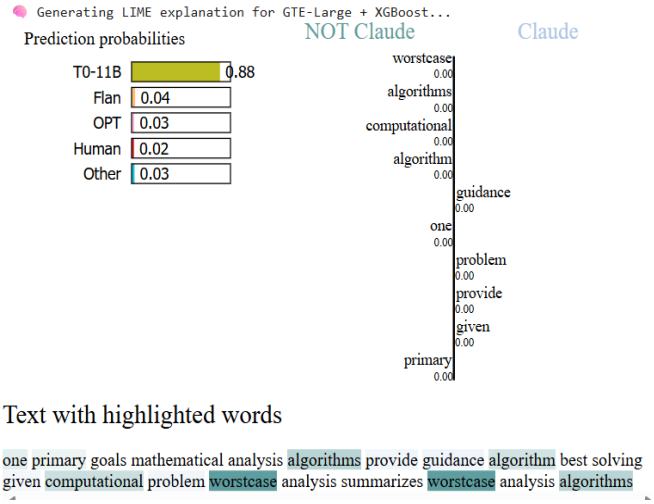
(a) D-BERT



(b) MPNet



(c) E5-base



(d) GTE-Large

Figure A2. LIME explanations

In more detail, the LIME interpretations for each embedding model can be described as follows:

DistilBERT (D-BERT)

- Emphasizes "algorithm" and "worst-case analysis" heavily across workouts.
- Shows that DistilBERT is a type of keyword-driven model, likely to have weights spread across multiple tokens and provide less focused explanations.

MPNet

- Emphasizes "algorithm" and "mathematical analysis" with stronger weights.
- Consistently captures contextual meaning more effectively, indicating that MPNet effectively encodes local terms and their surrounding context.

E5-base

- Gives weight not just to technical terms but also to function words such as "one" and "provide".
- Suggests that in some cases it does more linguistic work than what would be expected given just the content words, which may flatten interpretability.

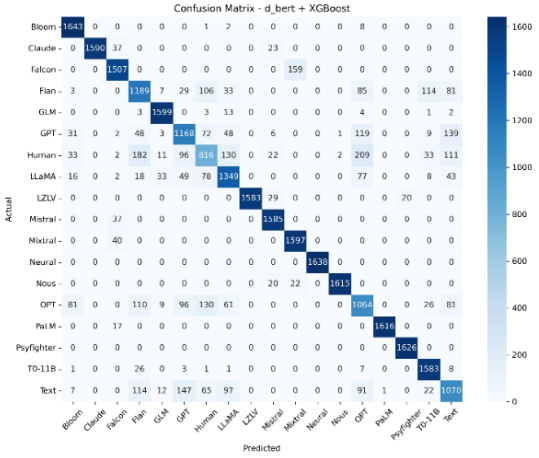
GTE-Large

- Produces the clearest, most focused highlights, especially on "algorithm", "worstcase", and "computational problem".
- It demonstrates sharper token attribution, making the explanations more understandable and believable compared to other models.

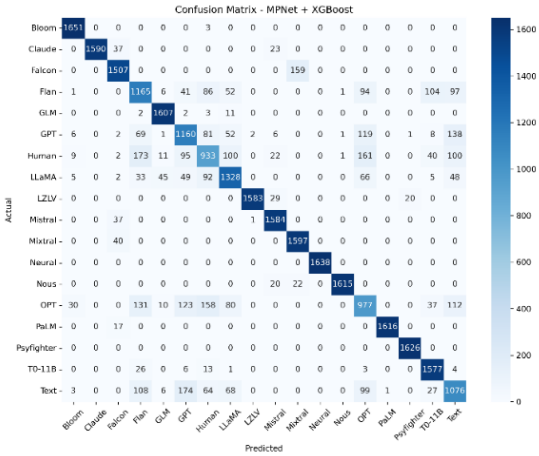
Overall, all four models enhance semantically meaningful tokens, with GTE-Large providing the most focused and precise explanations, MPNet providing a balance of contextual and keyword significance, DistilBERT spreading weight more diffusely, and E5-base extending saliency to functional words. This is consistent with our interpretation that LIME should be used in conjunction with SHAP to investigate the model-specific linguistic focus at the token level.

APPENDIX C. Full Confusion Matrix Visualizations

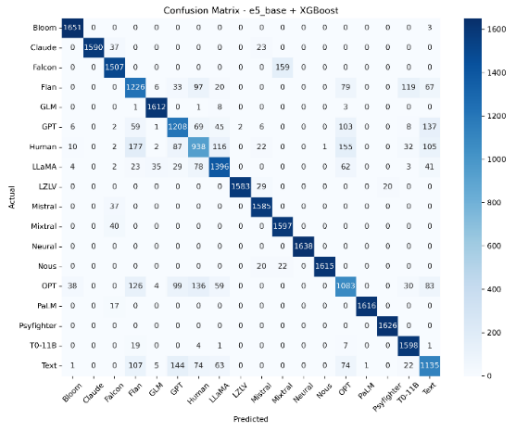
Figure A3 includes the complete confusion matrices for DistilBERT, MPNet, E5-base, and GTE-Large models, shown below. These provide the class-wise distribution of predictions beyond the summary discussed in Section 4.4.



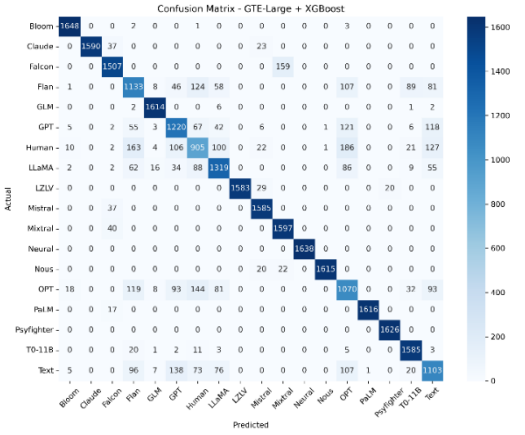
(a) DistilBERT + XGBoost



(b) MPNet + XGBoost



(c) E5-base + XGBoost



(d) GTE-Large + XGBoost

Figure A3. Confusion matrix

Appendix D. DistilRoBERTa Baseline Classification Report

This appendix contains the full experimental results for the fine-tuned DistilRoBERTa model, used as a baseline reference against the proposed embedding-based XGBoost framework. Only summary indicators are mentioned in the Results section, while full numerical outputs are documented here for transparency and reproducibility.

1. Baseline Performance Comparison

Table A1. Baseline performance comparison of different embedding–classifier combinations with inference time and explainability

Model	Accuracy	Precision	Recall	F1-Score	Inference Time	Explainability
E5-base + XGBoost	**0.89**	**0.88**	**0.89**	**0.88**	**8 ms**	**Full (SHAP + LIME)**
DistilRoBERTa (Fine-tuned)	0.88	0.88	0.88	0.879	34 ms	Limited
D-BERT + XGBoost	0.87	0.87	0.87	0.87	8 ms	Full
MPNet + XGBoost	0.87	0.87	0.87	0.87	8 ms	Full
GTE-Large + XGBoost	0.88	0.88	0.88	0.88	8 ms	Full

As shown in Table A1, the baseline performance of different embedding–classifier combinations is compared across multiple evaluation metrics, including accuracy, F1-score, inference time, and explainability.

Table A2. Class-wise classification performance of the fine-tuned DistilRoBERTa model on the 18-class dataset

Class	Precision	Recall	F1-score	Support
Bloom	0.9254	0.9746	0.9494	1654
Claude	0.9975	0.9636	0.9803	1650
Falcon	0.9262	0.9046	0.9153	1666
Flan	0.7021	0.6897	0.6959	1647
GLM	0.9019	0.9335	0.9174	1625
GPT	0.7099	0.7789	0.7428	1646
Human	0.7584	0.6709	0.7120	1647
LLaMA	0.8029	0.7693	0.7857	1673
LZLV	0.9757	0.9822	0.9789	1632
Mistral	0.9384	0.9772	0.9574	1622
Mixtral	0.8972	0.9756	0.9347	1637
Neural	0.9994	1.0000	0.9997	1638
Nous	0.9896	0.9747	0.9821	1657
OPT	0.7764	0.7352	0.7553	1658
PaLM	0.9927	1.0000	0.9963	1633
Psyfighter	0.9987	0.9803	0.9894	1626
T0-11B	0.7830	0.8920	0.8340	1630
Text	0.7549	0.6439	0.6950	1626
Accuracy 0.8801 29567				
Macro Avg	0.8795	0.8803	0.8790	29567
Weighted Avg	0.8793	0.8801	0.8788	29567

This benchmark confirms that our hybrid embedding-plus-

XGBoost framework provides an advantageous balance between accuracy, inference speed, and explainability, compared to a fine-tuned transformer baseline.

2. Full DistilRoBERTa Classification Report (18-Class Output)

Table A2 presents the complete classification results for the fine-tuned DistilRoBERTa model are provided below. These results demonstrate the class-level performance with respect to precision, recall, F1-score and support.