

Optimization of BERT, RoBERTa, and DeBERTa with XGBoost for Explainable Email Spam and Phishing Detection: A Comparative Analysis

Ugwueze Walter Oluchukwu and Nurudeen Oladehinbo Salau
University of the West of Scotland
Paisley, United Kingdom

Abstract—Email spam and phishing detection remains a challenging cybersecurity problem following the ability of malicious messages to exhibit highly contextual, deceptive, and rapidly changing linguistic characteristics. Conventional machine-learning approaches commonly depend on handcrafted features, whereas Transformer-based models provide richer contextual representations but can be computationally demanding and difficult to interpret. This study presents a comparative analysis of BERT, RoBERTa, and DeBERTa Transformer representations combined with XGBoost for email spam and phishing detection. The study first evaluates standalone Transformer classifiers and corresponding Transformer–XGBoost configurations, using the same controlled experimental framework and held-out evaluation data. The experiments use a controlled subset derived from an integrated corpus containing the Enron, CEAS 2008, and SpamAssassin email datasets. The comparative analysis is extended with DeBERTa-v3-base to examine whether a newer Transformer representation provides additional benefit when coupled with XGBoost. Results show that the DeBERTa–XGBoost configuration achieved 99.50% accuracy, 97.78% malicious-class recall, and 97.24% F1 on the held-out test set, compared with 99.40% accuracy, 95.56% recall, and 96.63% F1 for standalone DeBERTa. This represents a modest improvement, with false negatives reduced from four to two at the cost of one additional false positive. The earlier BERT and RoBERTa experiments provide comparative evidence that the benefit of adding XGBoost is not uniform across Transformer architectures. The study also examines computational cost by separating Transformer representation extraction from downstream XGBoost optimisation. This separation makes classifier training and optimisation inexpensive once fixed embeddings have been generated, although it does not establish lower end-to-end Transformer inference cost. Finally, SHAP and LIME are used to explain model decisions. SHAP identifies influential latent embedding dimensions, while LIME provides local textual explanations for true-positive, true-negative, false-positive, and false-negative cases. The LIME analysis reproduced the model predictions but showed limited stability in the exact ranking of influential textual features. The findings provide a comparative assessment of Transformer–XGBoost combinations and demonstrate the complementary value and limitations of latent-space and token-level explainability techniques.

Index Terms—email spam detection, phishing detection, BERT, RoBERTa, DeBERTa, XGBoost, Transformer embeddings, explainable artificial intelligence,

SHAP, LIME, cybersecurity

I. INTRODUCTION

Email remains one of the most widely used channels for communication and information exchange across organisations, government, education, healthcare, and personal environments. Its accessibility and scale, however, also make it an important medium for cyberattacks. Spam, phishing, spear-phishing, malware distribution, and related social-engineering attacks continue to present significant security challenges because malicious messages can be crafted to resemble legitimate communication while exploiting contextual, linguistic, and behavioural cues.

Traditional email filtering approaches have commonly relied on manually defined rules, blacklists, heuristic analysis, and handcrafted machine learning features. Although such approaches can be effective for known patterns, their dependence on predefined characteristics can limit their ability to recognise previously unseen or semantically deceptive messages. Machine-learning approaches improve adaptability by learning patterns from data, while deep learning methods can learn more complex representations directly from email content.

Transformer-based language models have further advanced text classification by learning contextual representations from sequences of tokens. In particular, BERT and RoBERTa have demonstrated the ability to capture semantic relationships within textual data and have consequently become important architectures for natural-language-based email classification. Nevertheless, Transformer models are computationally intensive and are often used as end-to-end classifiers, coupling contextual representation learning with the final classification stage. Their high-dimensional latent representations can also make individual decisions difficult to interpret, which is particularly problematic in cybersecurity applications where analysts may need to understand why an email was classified as malicious or legitimate.

These limitations motivate investigation of a modular classification strategy in which contextual representation learning is separated from downstream classification. In such an arrangement, a Transformer generates semantic representations and a separate classifier operates on the

resulting fixed-dimensional vectors. XGBoost is particularly suitable for this role because it provides a scalable gradient-boosted tree classifier that can be trained and optimised independently once the Transformer representations have been generated. The separation also creates an opportunity to compare different Transformer representations under a common downstream classification framework rather than assuming that the native Transformer classification head is optimal.

Explainability is a second important consideration. High-dimensional Transformer representations are not directly human-readable, and a high classification score alone does not reveal which aspects of a decision were important. Explainable Artificial Intelligence (XAI) techniques can therefore provide complementary evidence about model behaviour. SHAP can quantify the contribution of features used by a tree-based classifier, making it suitable for analysing influential dimensions of the Transformer embedding space. LIME can provide local explanations by identifying textual features that influence individual predictions. Using both methods makes it possible to investigate model behaviour at representation and text levels.

This study therefore focuses on the comparative optimisation of BERT, RoBERTa, and DeBERTa representations with XGBoost for email spam and phishing detection. The investigation is not based on the assumption that XGBoost will improve every Transformer model. Instead, it evaluates the standalone and Transformer-XGBoost configurations empirically and examines whether the effect of the downstream classifier differs across Transformer architectures. DeBERTa-v3-base is included as a later Transformer extension to the original BERT and RoBERTa experiments, allowing the study to assess whether a newer contextual representation provides additional benefit when combined with XGBoost.

The experimental investigation is organised around three connected considerations: classification performance, computational cost, and explainability. Classification performance is assessed using standard metrics, with particular attention to malicious-class recall and false negatives because missed malicious emails represent an important security risk. Computational analysis distinguishes the cost of generating Transformer representations from the cost of training and optimising the downstream XGBoost classifier. This distinction is important because separating the stages does not remove the need for Transformer computation when processing new emails. Finally, SHAP and LIME are used to examine the evidence underlying hybrid predictions, including representative true-positive, true-negative, false-positive, and false-negative cases.

A. Problem Statement

Despite advances in machine learning and Transformer-based natural language processing, reliable email spam and phishing detection remains challenging. Traditional

approaches can be limited by handcrafted features, while Transformer-based systems can require substantial computational resources and can be difficult to interpret. Furthermore, the use of a Transformer as an end-to-end classifier does not establish whether its learned contextual representation could be effectively reused by a separate classifier or whether such separation would make downstream experimentation more efficient.

A further limitation concerns explainability. A classifier may distinguish malicious from legitimate emails accurately while providing little evidence about the characteristics that influenced individual decisions. This is particularly important for false positives and false negatives, where security analysts need to understand the basis of an incorrect or potentially risky classification.

The problem addressed in this study is therefore the need to systematically evaluate whether contextual representations produced by different Transformer architectures can be effectively combined with XGBoost, whether the resulting combinations provide measurable classification benefits, what computational benefit is obtained from separating representation generation from downstream classifier optimisation, and how the resulting decisions can be investigated using complementary latent-space and text-level explanation methods.

B. Research Gap

The study addresses two interconnected research gaps.

First, Transformer-based email classifiers commonly combine computationally intensive representation learning with downstream classification. There is a need to investigate whether these functions can be separated and whether fixed Transformer representations can be reused for efficient downstream classifier optimisation. Importantly, the relevant computational question is not whether XGBoost eliminates the computational cost of the Transformer; the Transformer remains necessary to generate representations for new messages. Rather, the question is whether classifier experimentation can be performed more efficiently after the expensive representation-generation stage has been completed.

Second, the high-dimensional representations produced by Transformer models are difficult to interpret directly. Existing performance comparisons alone do not explain the evidence underlying individual decisions. A complementary explainability strategy is therefore required to investigate both the latent representation components used by the downstream classifier and the textual evidence associated with particular predictions. This study addresses this gap through SHAP-based latent representation analysis and LIME-based local text analysis.

C. Aim of the Study

The aim of this study is to comparatively evaluate the optimisation of BERT, RoBERTa, and DeBERTa Transformer representations with XGBoost for explainable email

spam and phishing detection, with particular emphasis on classification performance, computational cost, and model explainability.

D. Research Objectives

The study pursues the following objectives:

- 1) To prepare and evaluate a controlled email benchmark derived from the Enron, CEAS 2008, and SpamAssassin datasets using a consistent preprocessing and evaluation procedure.
- 2) To evaluate BERT, RoBERTa, and DeBERTa as Transformer-based contextual representation and classification models for email spam and phishing detection.
- 3) To combine Transformer-generated representations with XGBoost and compare the resulting Transformer-XGBoost configurations with their corresponding standalone Transformer models.
- 4) To optimise the downstream XGBoost classifier and determine whether changes in classifier configuration produce measurable differences in classification performance.
- 5) To investigate the computational characteristics of separating Transformer representation generation from downstream XGBoost training and optimisation.
- 6) To apply SHAP and LIME as complementary explainability techniques for analysing latent representation-level and local text-level evidence underlying model predictions.
- 7) To examine representative classification errors and assess the stability of local LIME explanations.

E. Research Questions

The study is guided by the following research questions:

- **RQ1:** How do BERT, RoBERTa, and DeBERTa-v3-base compare on the controlled email spam and phishing detection benchmark?
- **RQ2:** Does replacing the native Transformer classification stage with XGBoost improve held-out detection performance, and is the effect consistent across the three Transformer architectures?
- **RQ3:** What computational benefit is obtained by separating Transformer representation generation from downstream XGBoost classifier training and optimisation?
- **RQ4:** What latent and textual signals contribute to Transformer-XGBoost predictions, and how stable are the local LIME explanations?

F. Contributions of the Study

The study makes four principal contributions. First, it provides a controlled comparative evaluation of BERT, RoBERTa, and DeBERTa in both standalone and Transformer-XGBoost configurations. Second, it evaluates the practical effect of downstream XGBoost optimisation

rather than assuming that a separate classifier necessarily improves Transformer performance. Third, it distinguishes downstream classifier optimisation cost from the computational cost of Transformer representation generation, providing a more precise interpretation of computational efficiency. Fourth, it combines SHAP and LIME to provide complementary explanations at the latent representation and local textual levels, including analysis of both correct predictions and classification errors.

The resulting analysis is intended to provide evidence not only about which configuration performs well, but also about how the classification behaviour can be examined and what limitations remain when interpreting high-dimensional Transformer representations and local text explanations.

II. RELATED WORK

A. Conventional Machine Learning for Email Security

Early and conventional approaches to email spam and phishing detection typically relied on manually engineered lexical, structural, URL, and metadata features. Such approaches can provide effective discrimination when the characteristics associated with malicious messages are sufficiently stable, but their performance depends strongly on the quality and coverage of the engineered features. The literature reviewed in the underlying MSc study reports the use of machine-learning and natural-language-processing methods for phishing email detection, while also identifying the limitations of approaches that depend heavily on handcrafted representations [5]–[7].

Deep-learning approaches reduced the dependence on manual feature engineering by learning representations directly from email content. Studies reviewed in the dissertation reported improved spam and phishing classification using deep neural networks and learned representations [8]–[10]. However, these approaches do not necessarily provide contextual representations at the level achieved by modern Transformer language models, and computational cost and model interpretability remain recurring concerns.

B. Transformer-Based Email Classification

The introduction of Transformer-based language models substantially changed the representation of natural-language data. BERT introduced bidirectional contextual pre-training and demonstrated the effectiveness of Transformer representations across downstream language tasks [1]. RoBERTa subsequently revisited the BERT pre-training strategy, including changes to training data and optimisation, and demonstrated improved language-representation performance under its revised training procedure [2]. DeBERTa further extended the Transformer architecture through disentangled content and position representations and an enhanced mask decoder [3].

The relevance of these models to email security is that spam and phishing messages often depend on contextual relationships rather than isolated keywords. The literature

reviewed in the MSc study includes Transformer applications to phishing-related language analysis and reports evidence that Transformer models can capture persuasion and contextual characteristics in phishing emails. At the same time, Transformer-based systems remain computationally demanding and their learned representations are difficult to interpret directly.

The present study extends this line of work by evaluating three Transformer architectures—BERT, RoBERTa, and DeBERTa-v3-base—under a common experimental framework. Rather than treating the Transformer architecture as the only determinant of classification performance, the study separately evaluates the contextual representation and the downstream classification stage.

C. Transformer–XGBoost Hybrid Classification

Gradient-boosted decision trees provide an alternative classification layer that can operate on fixed-dimensional representations. XGBoost is designed for scalable gradient-boosted tree learning and supports efficient optimisation and parallel computation [4]. In a hybrid Transformer–XGBoost arrangement, the Transformer acts as a semantic encoder and XGBoost operates on the resulting embedding rather than replacing the Transformer itself.

This distinction is important because the hybrid approach has two separate computational stages:

$$\text{Email} \rightarrow \text{Transformer} \rightarrow \text{Embedding} \rightarrow \text{XGBoost} \rightarrow \text{Prediction} \quad (1)$$

The literature reviewed in the MSc study includes work investigating Transformer–XGBoost combinations for phishing detection. Mishra *et al.* reported a BERT–XGBoost framework for email phishing detection, while Vinnela and Chhabra investigated hybrid BERT, XGBoost, and Random Forest approaches for spam detection. These studies support the feasibility of combining contextual Transformer representations with tree-based classifiers, but they do not establish that XGBoost will consistently outperform a Transformer’s native classification stage across different Transformer architectures.

This distinction motivates the comparative design of the present study. BERT, RoBERTa, and DeBERTa are each evaluated with and without the downstream XGBoost stage. The objective is therefore empirical comparison, rather than an assumption that hybridisation necessarily produces a performance improvement.

D. Explainable Artificial Intelligence for Email Security

High predictive performance does not by itself explain why a model assigns a particular class to an email. This is especially relevant in cybersecurity, where analysts may need to inspect the evidence associated with a malicious classification or understand why a legitimate message was incorrectly flagged.

SHAP (SHapley Additive exPlanations) provides a feature-attribution framework based on Shapley-value

reasoning and can be applied to tree-based classifiers to quantify the contribution of individual input features [11]. In the present study, SHAP is applied to the 768-dimensional DeBERTa representation used by XGBoost. This provides a representation-level view of which latent dimensions contribute most strongly to the downstream classifier. These dimensions must not, however, be interpreted as direct human-readable concepts: a dimension with a large SHAP value cannot by itself be labelled as representing “phishing”, “malicious links”, or another specific semantic category.

LIME (Local Interpretable Model-agnostic Explanations) provides a different form of explanation by fitting an interpretable local surrogate around an individual prediction [12]. In this study, LIME is used to investigate textual evidence associated with representative true-positive, true-negative, false-positive, and false-negative predictions. This complements SHAP because the two methods operate at different levels: SHAP analyses the latent representation supplied to XGBoost, whereas LIME examines the original email text.

Recent work reviewed in the MSc dissertation has also explored explainable phishing detection using hybrid machine learning and XAI, BERT-based explainable approaches, boosting with SHAP, and explainable analysis of misclassifications. However, the reviewed studies identify a continuing need for approaches that combine contextual Transformer representations, downstream classification, and complementary explainability within one experimental framework.

E. Research Positioning

The literature establishes three relevant directions. First, conventional machine-learning and deep-learning approaches demonstrate the feasibility of automated email spam and phishing detection but can depend on engineered features or provide limited contextual understanding. Second, Transformer models provide richer contextual representations, while hybrid Transformer–XGBoost approaches demonstrate that these representations can be passed to a separate tree-based classifier. Third, XAI methods such as SHAP and LIME provide complementary mechanisms for investigating model behaviour.

However, three issues remain insufficiently resolved in the context of the present investigation. First, the comparative effect of the same downstream XGBoost strategy across BERT, RoBERTa, and DeBERTa has not been established by the experimental literature considered here. Second, computational benefit must be distinguished carefully from overall end-to-end computational cost: separating the classifier can make downstream optimisation inexpensive after embeddings are generated, but it does not remove the cost of Transformer representation generation. Third, explainable analysis should consider both latent representation-level evidence and human-readable textual evidence, while also examining the stability of local explanations.

The present study addresses these issues through a controlled comparative evaluation of standalone BERT, RoBERTa, and DeBERTa models and their corresponding XGBoost configurations, followed by computational analysis and a two-level SHAP–LIME explainability investigation. This positioning allows the study to determine whether hybridisation produces consistent benefits or whether its effect depends on the underlying Transformer representation.

III. MATERIALS AND METHODS

This study employed a controlled comparative experimental design to evaluate standalone Transformer classifiers and modular Transformer–XGBoost hybrid architectures for binary email security classification. The experimental pipeline was designed to separate contextual representation learning from downstream classifier optimisation and to evaluate classification performance, computational requirements, and explainability. Three Transformer families were considered: BERT, RoBERTa, and DeBERTa-v3-base [1]–[3]. The overall workflow was

Email Corpus → Cleaning and Normalisation → Transformer Fine-tuning → Contextual Embeddings → XGBoost Model Prediction

A. Dataset and Corpus Construction

The experimental corpus was constructed from three established email collections: the Enron Email Dataset, the CEAS 2008 Spam Dataset, and the SpamAssassin Public Corpus. The three sources were selected to provide a mixture of authentic corporate communication and benchmark spam/phishing content. Following integration and duplicate removal, the master corpus contained 258,656 unique email records. The source contributions were 215,638 messages from Enron, 39,039 from CEAS 2008, and 3,979 from SpamAssassin.

A unified schema was used to support consistent downstream processing. The principal fields were `dataset`, `subject`, `body`, `email_text`, `label`, and `cleaned_email_text`. The `email_text` field represented the combined subject and body, while `cleaned_email_text` was the normalised representation supplied to the Transformer models. The binary label was harmonised such that class 0 represented legitimate (ham) email and class 1 represented malicious, spam, or phishing email.

The original integrated collection contained 493,237 records. Duplicate detection removed 234,581 records, producing the final 258,656-record corpus. The resulting master dataset contained 7,969 missing values, concentrated primarily in the subject field. Importantly, the fields used directly by the modelling pipeline, including the cleaned text and binary label, were retained for the controlled modelling experiments.

B. Text Preprocessing and Normalisation

The data-engineering pipeline applied deterministic text-cleaning and normalisation before tokenisation. The

TABLE I
CONTROLLED EXPERIMENTAL DATASET

Subset	Total	Class 0	Class 1
Training	5,000	4,552	448
Validation	1,000	910	90
Held-out test	1,000	910	90

operations included removal of HTML tags, decoding of HTML entities, removal of non-printable control characters, whitespace normalisation, and Unicode normalisation using NFKC. These operations were intended to reduce formatting noise while retaining the textual content required for contextual modelling.

Token-length analysis showed substantial variation in email length. The maximum observed token length was 1,123,760 tokens, with a mean of 495.41 and a median of 211 tokens. A total of 61,656 messages (23.84%) exceeded 512 tokens. A maximum sequence length of 512 tokens was therefore adopted for the Transformer experiments. Messages exceeding this limit were handled according to the tokenizer’s maximum-length truncation behaviour, while shorter messages were padded to the required length for processing.

C. Experimental Splitting and Class Distribution

The full processed corpus was stratified into training, validation, and testing partitions of 70%, 15%, and 15%, respectively. For the controlled Transformer experiments, a fixed subset consisting of 5,000 training messages, 1,000 validation messages, and 1,000 held-out test messages was used. Random state 42 was applied to maintain reproducibility.

The controlled subset was intentionally imbalanced. The training set contained 4,552 class-0 and 448 class-1 messages, while both the validation and held-out test sets contained 910 class-0 and 90 class-1 messages. Because the malicious class was substantially smaller, class weighting was used during Transformer fine-tuning. The held-out test set was reserved for final evaluation and was not used for model selection or hyperparameter optimisation.

D. Transformer Models

Three pretrained Transformer architectures were evaluated. BERT used the `bert-base-uncased` checkpoint, RoBERTa used `roberta-base`, and the additional controlled experiment used `microsoft/deberta-v3-base`. BERT and RoBERTa provide established bidirectional contextual encoders, whereas DeBERTa extends this family through disentangled attention and enhanced decoding mechanisms [1]–[3].

For BERT and RoBERTa, the fine-tuning configuration used a maximum sequence length of 512 tokens, batch size 8, three training epochs, learning rate 2×10^{-5} , AdamW optimisation, weighted cross-entropy loss, and gradient clipping with a maximum norm of 1.0. BERT used 10% warm-up of the training steps, while the documented

TABLE II
TRANSFORMER FINE-TUNING CONFIGURATION

Parameter	Configuration
BERT model	bert-base-uncased
RoBERTa model	roberta-base
DeBERTa model	microsoft/deberta-v3-base
Maximum sequence length	512 tokens
Batch size	8
Epochs	3
Learning rate	2×10^{-5}
Weight decay	0.01
Optimiser	AdamW
Loss	Weighted cross-entropy
Gradient clipping	Maximum norm = 1.0
Warm-up	10% for BERT/DeBERTa; 0 documented for RoBERTa
Class weights	0.5492 (class 0), 5.5804 (class 1)
Random state	42

RoBERTa configuration used zero warm-up steps. The class weights were 0.5492 for class 0 and 5.5804 for class 1.

The DeBERTa-v3-base experiment used the same principal fine-tuning scale: batch size 8, three epochs, learning rate 2×10^{-5} , weight decay 0.01, AdamW optimisation, weighted cross-entropy, gradient clipping, and 10% warm-up. The best DeBERTa checkpoint was selected using validation F1 and occurred at epoch 2. The selected model produced a 768-dimensional representation for each email.

E. Contextual Embedding Extraction

After fine-tuning, the Transformer encoders were used as representation learners for the hybrid experiments. For BERT, the representation at the first token position ([CLS]) from the final hidden state was extracted. The corresponding contextual representation strategy was applied to the DeBERTa-v3-base experiment using its first-token hidden-state representation. The resulting vectors were treated as fixed numerical features for the downstream XGBoost classifier.

The controlled embedding sets comprised 5,000 training vectors, 1,000 validation vectors, and 1,000 test vectors, each with 768 dimensions. Thus, the downstream classifier received a fixed-dimensional representation rather than the original token sequence. This separation also allowed multiple XGBoost configurations to be compared without regenerating the Transformer representations.

A logistic-regression classifier was additionally applied to the DeBERTa embeddings as a diagnostic baseline. This was not treated as one of the principal hybrid models; rather, it was used to assess whether substantial class-separating information was already present in the learned embedding space before tree-based optimisation.

F. XGBoost Downstream Classification

XGBoost was used as the downstream classifier because gradient-boosted decision trees can model nonlinear relationships among fixed numerical features and provide a

natural basis for post-hoc SHAP explanations [4], [11]. The hybrid architecture can therefore be expressed as

$$\mathbf{z} = f_T(x), \quad \hat{y} = f_{XGB}(\mathbf{z}),$$

where x is the cleaned email text, f_T is the fine-tuned Transformer, $\mathbf{z} \in \mathbb{R}^{768}$ is the extracted contextual representation, and f_{XGB} is the downstream XGBoost classifier.

The documented baseline XGBoost configuration for the original BERT/RoBERTa hybrid experiments used binary logistic prediction, log-loss evaluation, 300 estimators, maximum tree depth 6, learning rate 0.05, subsampling of 0.80, column subsampling of 0.80, random state 42, and all available processing jobs. The BERT-XGBoost and RoBERTa-XGBoost experiments used Transformer-generated embeddings as the input feature matrix.

For the controlled DeBERTa experiment, the XGBoost stage was optimised separately after embedding extraction. The baseline and four controlled variants represented shallower trees, deeper trees, increased regularisation, and a lower-learning-rate configuration with more estimators. The principal selected configuration was:

- objective: `binary:logistic`;
- evaluation metric: `log-loss`;
- estimators: 800;
- learning rate: 0.03;
- maximum depth: 6;
- minimum child weight: 1;
- subsample: 0.80;
- column subsampling: 0.80;
- gamma: 0;
- L1 regularisation: 0;
- L2 regularisation: 1;
- class imbalance weight: 10.160714;
- random state: 42; and
- tree method: `histogram-based training`.

The controlled XGBoost configurations were compared on the validation set. Model selection prioritised F1-score, followed by malicious-class recall and PR-AUC. The selected configuration was then evaluated once on the held-out test set.

G. Evaluation Metrics

Performance was evaluated using accuracy, precision, recall, F1-score, balanced accuracy, ROC-AUC, PR-AUC, Matthews correlation coefficient (MCC), Cohen’s kappa, and confusion matrices. These metrics were selected because the class distribution was imbalanced and because false negatives are particularly important in email security.

For binary classification, accuracy is defined as

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

Precision and recall are

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN},$$

and F1-score is

$$F_1 = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Balanced accuracy was used to give equal weight to the two class-specific recalls. ROC-AUC measured ranking performance across classification thresholds, while PR-AUC was emphasised because it is informative when the positive class is relatively uncommon. MCC and Cohen’s kappa were included as additional agreement measures that account for the class distribution.

In the security-oriented interpretation, class 1 corresponds to malicious email. Consequently, false negatives (malicious emails classified as legitimate) were explicitly examined alongside overall accuracy and F1.

H. Computational Cost Measurement

Computational cost was assessed by recording the practical cost of Transformer representation generation and by examining the modularity of the downstream XGBoost stage. The DeBERTa embedding extraction process was measured separately for the training, validation, and test representations. The total local CPU extraction time was approximately 264.55 minutes.

This measurement was interpreted as the dominant representation-generation cost of the hybrid pipeline. Once the fixed 768-dimensional embeddings had been produced, alternative XGBoost configurations could be trained and compared without rerunning the Transformer. This provides a practical modularity advantage for experimentation and classifier optimisation. However, the study did not conduct a matched end-to-end benchmark of inference latency, throughput, peak memory, or energy consumption. Therefore, the hybrid architecture is not claimed to have lower total end-to-end computational cost than standalone Transformer inference.

I. Explainability Framework

Explainability was evaluated at two complementary levels. SHAP was applied to the XGBoost classifier to quantify the contribution of the latent Transformer dimensions to individual predictions and to derive global importance rankings [11]. Mean absolute SHAP values were used as the principal global importance measure. The analysis examined the most influential latent dimensions and selected representative true positive, true negative, false positive, and false negative cases.

Because the XGBoost model operates on latent 768-dimensional representations, a SHAP dimension is treated as a learned numerical feature rather than as a direct human-interpretable word or semantic concept. Consequently, the results are interpreted as evidence about which latent representation dimensions influence the classifier rather than as direct statements that a particular dimension represents a particular phishing phrase.

LIME was used as a complementary text-level local explanation method [12]. LIME perturbs the input text

and fits a local surrogate model around the instance being explained. The experiments used the same maximum sequence length of 512 tokens and evaluated representative true-positive, true-negative, false-positive, and false-negative cases. Initial explanations used 120 perturbation samples per case, followed by a 500-sample analysis to assess explanation robustness. Twenty local features were requested in each explanation.

The LIME analysis evaluated both prediction preservation and feature-ranking stability. Prediction probabilities remained aligned with the locked hybrid-model outputs when the perturbation budget was increased, whereas the exact rankings of the selected textual features changed substantially. Accordingly, LIME results were treated as local illustrative evidence rather than as a stable global feature ranking.

J. Experimental Protocol and Test-Set Isolation

The experimental protocol maintained a strict separation between model selection and final evaluation. Transformer models were fine-tuned using the training data, with validation performance used for checkpoint selection. The XGBoost variants were subsequently compared using the validation set. Only after the Transformer checkpoint and downstream XGBoost configuration had been selected was the held-out test set used for final comparison.

The same controlled test partition was used for standalone DeBERTa and DeBERTa–XGBoost evaluation. This paired evaluation allowed changes in the error profile to be assessed directly, including changes in false positives, false negatives, recall, and balanced accuracy. The BERT and RoBERTa experiments were likewise evaluated under the controlled experimental framework, allowing the effect of downstream XGBoost classification to be compared across Transformer families.

The complete methodology therefore supports three linked analyses: (i) whether downstream XGBoost improves Transformer classification performance, (ii) whether the modular architecture provides a practical computational advantage for repeated classifier optimisation, and (iii) whether SHAP and LIME provide complementary explanations at latent and text levels.

IV. EXPERIMENTAL RESULTS

This section presents the empirical results obtained from the controlled evaluation of the standalone Transformer and Transformer–XGBoost architectures. The results are reported in two stages. First, the previously evaluated BERT and RoBERTa models are compared with their corresponding hybrid classifiers. Second, the additional DeBERTa-v3-base experiment is examined in greater detail, including the controlled XGBoost optimisation and the final held-out test comparison. The interpretation emphasises malicious-class recall and false-negative behaviour in addition to aggregate classification performance.

A. Overall Comparison of Transformer and Hybrid Models

Table III presents the principal held-out test results for the six evaluated architectures. All models achieved accuracy above 99% on the controlled test set. The standalone BERT model achieved 99.40% accuracy, while RoBERTa achieved 99.30%. Their corresponding Transformer-XGBoost hybrids achieved 99.20% accuracy.

The DeBERTa-v3-base model achieved 99.40% accuracy, matching BERT, while also producing a higher ROC-AUC of 99.92% and PR-AUC of 99.38%. The DeBERTa-XGBoost hybrid produced the highest observed accuracy at 99.50%, together with the highest recall, F1-score, balanced accuracy, ROC-AUC, PR-AUC, and MCC among the evaluated configurations.

The results show that the effect of XGBoost was model-dependent. For BERT and RoBERTa, replacing the native Transformer classification stage with the downstream XGBoost classifier reduced the principal threshold-based classification measures. In contrast, the DeBERTa hybrid produced a modest improvement over standalone DeBERTa. Therefore, the results do not support the claim that XGBoost universally improves Transformer-based email classification. Instead, the evidence indicates that the value of the hybrid architecture depends on the representation produced by the upstream Transformer and the behaviour of the downstream decision boundary.

B. BERT and RoBERTa Baseline Results

The standalone BERT model produced 99.40% accuracy, 97.73% precision, 95.56% recall, and 96.63% F1-score. RoBERTa produced 99.30% accuracy, 96.63% precision, 95.56% recall, and 96.09% F1-score. The balanced accuracy values were 97.67% and 97.61%, respectively.

These results establish a strong baseline for the hybrid comparisons. Both models demonstrated high discriminatory performance despite the imbalanced class distribution. BERT obtained a small advantage over RoBERTa in precision, F1-score, MCC, and Cohen’s kappa, while their ROC-AUC values were both 99.84%.

C. BERT-XGBoost and RoBERTa-XGBoost Results

The BERT-XGBoost model achieved 99.20% accuracy, 96.59% precision, 94.44% recall, and 95.51% F1-score. Its balanced accuracy was 97.06%, ROC-AUC was 99.17%, PR-AUC was 98.11%, and MCC was 95.07%.

The RoBERTa-XGBoost model produced the same threshold-based accuracy, precision, recall, F1-score, balanced accuracy, and MCC. Its ROC-AUC was 99.76%, while PR-AUC was 98.34%. Thus, although the RoBERTa hybrid maintained very strong ranking performance, neither hybrid exceeded its standalone Transformer counterpart on the principal classification measures.

The BERT and RoBERTa results are therefore important to the comparative research question: the downstream tree-based classifier was technically feasible but did not automatically translate into improved predictive performance. This

provides a useful counterpoint to approaches that assume hybridisation is beneficial solely because it combines two model families.

D. DeBERTa Standalone Versus DeBERTa-XGBoost

The focused DeBERTa experiment produced a different outcome. Standalone DeBERTa achieved 99.40% accuracy, 97.73% precision, 95.56% recall, and 96.63% F1-score. The hybrid increased accuracy to 99.50%, recall to 97.78%, and F1-score to 97.24%. Balanced accuracy increased from 97.67% to 98.72%, ROC-AUC from 99.92% to 99.98%, and PR-AUC from 99.38% to 99.78%. MCC increased from 96.31% to 96.96%.

The improvement was not uniform across metrics. Precision decreased from 97.73% to 96.70%, representing an increase in false-positive decisions. The principal improvement was therefore increased sensitivity to the malicious class rather than a general improvement in every metric.

The confusion matrices provide a direct security-oriented interpretation. Standalone DeBERTa produced 908 true negatives, 2 false positives, 4 false negatives, and 86 true positives. The hybrid produced 907 true negatives, 3 false positives, 2 false negatives, and 88 true positives. Thus, the hybrid detected two additional malicious emails while producing one additional false alarm.

From a security perspective, the reduction in false negatives is important because false-negative cases correspond to malicious emails being classified as legitimate. The observed reduction from four to two represents a 50% reduction in false negatives on this controlled test set. However, this improvement was accompanied by an increase in false positives from two to three. The result is therefore best characterised as a modest shift toward greater malicious-class sensitivity rather than an across-the-board performance improvement.

E. DeBERTa Embedding Diagnostic

Before downstream XGBoost optimisation, a logistic-regression classifier was used as a diagnostic baseline on the fixed DeBERTa embeddings. On the validation set, logistic regression achieved 99.50% accuracy, 98.85% precision, 95.56% recall, 97.18% F1-score, 97.72% balanced accuracy, 99.57% ROC-AUC, 98.20% PR-AUC, and 96.92% MCC.

This result indicates that substantial class-separating information was already encoded in the 768-dimensional DeBERTa representation before XGBoost optimisation. The observation is consistent with the later finding that moderate changes to the XGBoost configuration produced only limited changes to thresholded validation predictions.

F. Controlled XGBoost Optimisation

Five XGBoost configurations were evaluated on the DeBERTa validation embeddings. The configurations represented the baseline, shallower trees, deeper trees, increased regularisation, and a lower-learning-rate model with more estimators.

TABLE III
HELD-OUT TEST PERFORMANCE ACROSS EVALUATED ARCHITECTURES

Model	Acc.	Prec.	Rec.	F1	Bal. Acc.	ROC-AUC	PR-AUC	MCC
BERT	99.40	97.73	95.56	96.63	97.67	99.84	98.79	96.31
RoBERTa	99.30	96.63	95.56	96.09	97.61	99.84	98.34	95.71
BERT + XGBoost	99.20	96.59	94.44	95.51	97.06	99.17	98.11	95.07
RoBERTa + XGBoost	99.20	96.59	94.44	95.51	97.06	99.76	98.34	95.07
DeBERTa-v3-base	99.40	97.73	95.56	96.63	97.67	99.92	99.38	96.31
DeBERTa-v3-base + XGBoost	99.50	96.70	97.78	97.24	98.72	99.98	99.78	96.96

TABLE IV
FOCUSED DEBERTA AND DEBERTA-XGBOOST COMPARISON

Metric	Standalone	Hybrid	Change (pp)
Accuracy	99.40	99.50	+0.10
Precision	97.73	96.70	-1.02
Recall	95.56	97.78	+2.22
F1	96.63	97.24	+0.61
Balanced Accuracy	97.67	98.72	+1.06
ROC-AUC	99.92	99.98	+0.06
PR-AUC	99.38	99.78	+0.40
MCC	96.31	96.96	+0.66

TABLE V
CONFUSION-MATRIX COMPARISON FOR DEBERTA MODELS

Model	TN	FP	FN	TP
Standalone DeBERTa	908	2	4	86
DeBERTa + XGBoost	907	3	2	88

All five configurations produced identical thresholded validation classifications: 99.30% accuracy, 100.00% precision, 92.22% recall, 95.95% F1-score, and 96.11% balanced accuracy. The common confusion matrix was

$$\begin{bmatrix} 910 & 0 \\ 7 & 83 \end{bmatrix}.$$

Although the thresholded predictions were identical, ranking performance varied slightly between configurations. ROC-AUC ranged from 99.76% to 99.87%, while PR-AUC ranged from 98.68% to 99.06%. The selected lower-learning-rate configuration achieved the highest ROC-AUC of 99.87% and PR-AUC of 99.06%.

The identical thresholded classifications indicate that moderate changes in tree complexity did not materially alter the decision labels in the controlled validation subset. The small changes in ROC-AUC and PR-AUC, however, show that the configurations did differ slightly in probability ranking. The lower-learning-rate model was consequently retained for final held-out evaluation because it achieved the strongest ranking metrics while preserving the same thresholded validation outcome.

G. ROC and Precision-Recall Analysis

The final DeBERTa comparison produced near-ceiling ROC and precision-recall performance for both models. The standalone DeBERTa model achieved a ROC-AUC of 99.92% and PR-AUC of 99.38%, while the DeBERTa-XGBoost hybrid achieved 99.98% and 99.78%, respectively. These results indicate that both models ranked malicious

TABLE VI
CONTROLLED XGBOOST VALIDATION RESULTS

Configuration	ROC-AUC	PR-AUC
Baseline	99.86	99.01
Shallower trees	99.86	99.00
Deeper trees	99.86	99.01
Regularised	99.76	98.68
Lower learning rate	99.87	99.06

and legitimate emails very effectively on the controlled held-out set, with the hybrid providing a small improvement in both ranking measures.

If the corresponding figures are included in the IEEE manuscript, the ROC and precision-recall curves should be generated from the locked held-out test predictions and probabilities so that the figures remain consistent with the reported AUC values.

H. Security-Oriented Error Analysis

The confusion-matrix results were examined further using representative true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) cases. The purpose was to determine whether the aggregate metrics concealed qualitatively different classification behaviour.

The representative TP was a CEAS email with the subject “Thousands of products”. Its content contained promotional material and an external domain and was correctly classified as malicious with a probability of approximately 0.999956. The representative TN was an Enron email with the subject “NBPL Ratebase Multiple / NBP Unit Price Comparison”. It contained long-form internal business analysis and was correctly classified as legitimate with a malicious probability of approximately 0.000050.

The representative FP was an Enron email labelled legitimate with the subject “Increase Sales.....”. Despite its dataset label, the message contained highly promotional and spam-like characteristics, including claims of substantially increasing sales, credit-card references, pre-approval language, and a call to click. The hybrid model assigned a malicious probability of approximately 0.999699. This case illustrates a distinction between the dataset-defined label and the content-level characteristics presented to the classifier; it should not, by itself, be treated as proof of a labelling error.

The representative FN was a CEAS email with the subject “rent, How to WIN \$6950 in Targeted Promo-

TABLE VII
REPRESENTATIVE DeBERTa-XGBoost ERROR-ANALYSIS CASES

Case	Index	True/Pred.	$P(y = 1)$
True positive	654	1/1	0.999956
True negative	977	0/0	0.000050
False positive	191	0/1	0.999699
False negative	289	1/0	0.003293

tions?”. The message contained promotional and giveaway indicators, including “FREE”, “MAXIMUM RETURNS”, external URLs, and unsubscribe-related material, but the model assigned a malicious probability of only approximately 0.003293. The associated latent analysis indicated competing evidence, with legitimate-oriented signals ultimately dominating the downstream decision.

The error analysis therefore complements the aggregate metrics. The false positive demonstrates that a legitimate-labelled message can exhibit strong malicious/spam-like textual characteristics, whereas the false negative demonstrates that overtly promotional or suspicious content can still be represented in a way that produces competing legitimate-oriented signals. These cases motivate the explainability analysis presented in the next section.

I. Summary of Experimental Findings

Three principal findings emerge from the experimental results. First, the standalone BERT and RoBERTa models performed extremely strongly, and their corresponding XGBoost hybrids did not improve the principal classification measures. Second, the DeBERTa-v3-base representation supported a modest but measurable improvement when coupled with the selected XGBoost classifier, particularly in malicious-class recall, F1-score, and balanced accuracy. Third, the controlled XGBoost optimisation produced identical thresholded validation predictions across all tested configurations, indicating that the learned DeBERTa representation was already highly discriminative and that XGBoost primarily provided limited decision-boundary refinement.

The strongest security-oriented result was the reduction in false negatives from four to two on the held-out DeBERTa test set, at the cost of one additional false positive. The following sections therefore examine the computational implications of the modular architecture and the interpretability of its decisions.

V. COMPUTATIONAL COST ANALYSIS

The computational analysis was designed to investigate whether separating Transformer representation learning from downstream XGBoost classification provides a practical computational advantage. The analysis therefore distinguishes between the cost of generating contextual representations and the cost of repeatedly training and optimising the downstream classifier. This distinction is important because the hybrid architecture still requires

TABLE VIII
MEASURED DeBERTa REPRESENTATION-EXTRACTION COST

Subset	Emails	Time (min)
Training	5,000	193.30
Validation	1,000	30.61
Held-out test	1,000	40.61
Total	7,000	264.55

the Transformer to process an email before XGBoost can make a prediction.

A. Measured Computational Components

The principal measured computational component was DeBERTa-v3-base representation generation. The 5,000 training, 1,000 validation, and 1,000 test emails were passed through the fine-tuned DeBERTa model to obtain fixed 768-dimensional representations. The complete embedding-generation process on the local CPU required approximately 264.55 minutes.

The extraction times were approximately 193.30 minutes for the training set, 30.61 minutes for the validation set, and 40.61 minutes for the held-out test set. The corresponding extraction throughputs were approximately 0.43, 0.54, and 0.41 emails per second, respectively. The variation reflects the practical processing characteristics of the three batches rather than a change in the underlying model architecture.

The results show that representation generation is the dominant measured component of the hybrid workflow. This is expected from the architecture: the Transformer performs the computationally intensive contextual encoding, whereas XGBoost operates on the resulting fixed-dimensional numerical vectors.

B. Downstream XGBoost Optimisation Cost

After representation extraction, the same fixed DeBERTa embeddings were reused for five controlled XGBoost experiments. The measured training time for an individual configuration was approximately 15.9–27.9 seconds. Consequently, changing tree depth, regularisation, or learning rate did not require the Transformer to process the email corpus again.

This creates an important computational separation:

$$T_{\text{hybrid,training}} \approx T_{\text{representation}} + T_{\text{XGBoost}},$$

where the measured representation-generation term was substantially larger than the downstream classifier-training term. Once the first term had been paid, additional XGBoost experiments could be performed rapidly.

The practical advantage is therefore not that XGBoost replaces the Transformer computational burden. Rather, the fixed representation creates a reusable interface between representation learning and downstream classification. For example, five alternative XGBoost configurations can be trained and compared without repeating the approximately 264.55-minute embedding-generation process.

TABLE IX
COMPUTATIONAL CHARACTERISTICS OF THE CONTROLLED HYBRID
WORKFLOW

Component	Measured Cost
DeBERTa train embeddings	193.30 min
DeBERTa validation embeddings	30.61 min
DeBERTa test embeddings	40.61 min
Total representation extraction	264.55 min
XGBoost configuration	15.9–27.9 s

C. Representation Reuse and Experimental Modularity

The measured results demonstrate a modular computational workflow. The Transformer can be treated as a representation generator whose output is a fixed 768-dimensional feature matrix. The downstream classifier can then be optimised independently over that matrix.

This modularity has particular value during experimentation. If a new XGBoost configuration is required, the expensive Transformer stage does not need to be repeated provided that the Transformer checkpoint and representations remain unchanged. The computational structure can therefore be represented as

$$\text{Email} \xrightarrow{\text{DeBERTa}} \mathbf{z}_{768} \xrightarrow{\text{XGBoost}} \hat{y},$$

with $\mathbf{z}_{768}$ stored and reused for downstream experiments.

The same principle also supports comparative evaluation. The fixed representation allows classifier changes to be studied while keeping the upstream semantic representation constant. This reduces a potential confounding factor in experiments where the objective is specifically to study the effect of the downstream classifier.

D. Inference-Time Considerations

The computational advantage observed during offline experimentation should not be interpreted as evidence that the complete hybrid system has lower per-email inference cost. For a new email, the Transformer must still generate its contextual representation before XGBoost can classify the resulting vector. Consequently, the expensive representation-generation stage remains part of the inference path.

The recorded standalone DeBERTa evaluation for the 1,000-email test set required approximately 1,743.93 seconds. An equivalent end-to-end timing measurement for the DeBERTa–XGBoost pipeline was not recorded under the same experimental conditions. It is therefore not possible from the available measurements to establish a statistically valid reduction in per-email latency, throughput, peak memory usage, or energy consumption.

Accordingly, the computational finding is deliberately limited to the measured workflow: XGBoost optimisation was inexpensive once the DeBERTa representations had been generated, whereas the Transformer representation stage remained computationally dominant.

E. Computational Cost and the Research Gap

The results partially address the computational-cost research gap identified in the study. They demonstrate experimentally that separating contextual representation generation from downstream classifier optimisation can make repeated classifier experimentation substantially cheaper in computational time after the representations have been generated.

However, the results do not establish that the complete hybrid architecture is cheaper than standalone DeBERTa for either training or deployment. In particular, the Transformer remains necessary for new inputs, and the study did not perform matched measurements of end-to-end inference latency, throughput, peak memory, or energy consumption.

The appropriate conclusion is therefore that the architecture provides a *computational modularity advantage*, rather than a demonstrated reduction in total end-to-end computational cost. This distinction prevents the measured reuse benefit from being overstated as a deployment-efficiency claim.

F. Implications for Practical Model Development

From a model-development perspective, the computational results support a two-stage workflow. The first stage is the relatively expensive learning or generation of contextual representations. The second stage is rapid experimentation with downstream classifiers using those fixed representations. This separation can be useful when comparing multiple tree-based configurations, conducting sensitivity analyses, or applying different post-hoc explainability procedures.

The controlled experiment illustrates this directly: five XGBoost configurations were evaluated without regenerating the DeBERTa embeddings, and their thresholded validation predictions were identical. The additional classifier experiments therefore incurred only seconds of downstream training time per configuration rather than another full Transformer representation-generation run.

This finding also helps explain the classification results reported in Section IV. Because the DeBERTa representation already contained strong class-separating information, moderate changes to XGBoost complexity produced limited changes to the final validation decisions. The primary role of XGBoost in this experiment was consequently decision-boundary refinement on a strong contextual representation rather than replacement of the representation-learning process.

G. Summary

The computational analysis produced three principal observations. First, DeBERTa representation generation dominated the measured computational workflow, requiring approximately 264.55 minutes for the controlled training, validation, and test representations on the local CPU. Second, once those 768-dimensional representations were available, XGBoost experiments were comparatively

TABLE X
TOP LATENT DIMENSIONS BY MEAN ABSOLUTE SHAP VALUE FOR
DeBERTa-XGBoost.

Latent Dimension	Mean SHAP
dim_440	1.169435
dim_129	0.852690
dim_587	0.559432
dim_230	0.476085
dim_451	0.417720
dim_555	0.408240
dim_157	0.365455
dim_510	0.352385
dim_659	0.320761
dim_303	0.309468

inexpensive, requiring approximately 15.9–27.9 seconds per controlled configuration. Third, the study did not demonstrate lower end-to-end training or inference cost for the hybrid architecture because matched deployment benchmarks were not performed.

Thus, the principal computational contribution of the hybrid design is **modularity and representation reuse**. The result is practically useful for repeated downstream classifier optimisation, but it should not be presented as evidence that Transformer-XGBoost inference is intrinsically faster or cheaper than standalone Transformer inference.

VI. EXPLAINABILITY ANALYSIS

Explainability was evaluated using two complementary approaches. SHAP was applied to the XGBoost classifier to quantify the contribution of the learned DeBERTa embedding dimensions to the hybrid prediction, while LIME was applied at the original text level to provide locally interpretable token- and word-level evidence. This combination was selected because the two methods operate at different representation levels: SHAP explains the downstream classifier in the latent embedding space, whereas LIME constructs a local surrogate around the original email text. The analysis therefore distinguishes latent-feature attribution from text-level illustration rather than treating the two explanations as interchangeable.

A. Global SHAP Analysis

The selected DeBERTa-XGBoost model represents each email using a 768-dimensional embedding obtained from the first-token hidden representation of the fine-tuned DeBERTa model. SHAP was subsequently used to explain the XGBoost decision function over these fixed representations. The global importance measure was the mean absolute SHAP value across the evaluated instances. This measure captures the average magnitude of a latent dimension’s contribution to the classifier output without cancelling positive and negative contributions.

Table X presents the ten most influential latent dimensions according to mean absolute SHAP value.

The results indicate that the hybrid classifier did not rely uniformly on all 768 dimensions. In particular, dim_440 had the largest average absolute contribution, followed by

dim_129 and dim_587. The concentration of importance in a subset of dimensions provides evidence that the downstream classifier exploited discriminative structure contained in the Transformer representation.

These dimensions should, however, be interpreted as learned latent numerical features rather than human-readable semantic concepts. A dimension such as dim_440 cannot be interpreted directly as “phishing,” “urgency,” “a malicious URL,” or any other individual textual property. The DeBERTa representation is distributed, and a semantic concept may be encoded across multiple dimensions. Consequently, SHAP provides a faithful attribution of the XGBoost stage while not, by itself, providing a direct linguistic explanation of the source email.

B. Local SHAP Error Analysis

To examine how the hybrid classifier behaved on individual predictions, four representative held-out test cases were analysed: a true positive (TP), a true negative (TN), a false positive (FP), and a false negative (FN). The corresponding hybrid probabilities were approximately 0.999956, 0.000050, 0.999699, and 0.003293, respectively. These cases were selected to cover both correct and erroneous decisions and therefore provide a more informative explanation analysis than examining only correctly classified emails.

For the TP case, the dominant contributing latent dimensions provided coherent evidence in favour of the malicious class, producing a probability of 0.999956. The TN case showed the opposite pattern: influential dimensions contributed in a direction supporting the legitimate class, resulting in a probability of only 0.000050 for the malicious class. These two examples illustrate how the same latent representation space can support opposite classification outcomes through different combinations of feature values and XGBoost decision paths.

The FP case is particularly informative from a security perspective. The legitimate-labelled email received a malicious probability of 0.999699. Its content contained strongly promotional and suspicious language, including claims of dramatically increasing sales, references to credit cards, pre-approval language, and a “click here” call to action. The SHAP explanation therefore indicates that the model detected a latent pattern strongly associated with malicious messages even though the assigned label was legitimate. This illustrates the ambiguity that can arise between learned model evidence and dataset labels; however, the case alone does not establish that the label is erroneous.

The FN case represents the most security-relevant error because a malicious email was assigned a malicious probability of only 0.003293. The message contained promotional and giveaway-oriented language together with external links and related call-to-action material. In this instance, the latent evidence supporting the legitimate class outweighed the evidence supporting the malicious class. The case demonstrates why error analysis remains necessary

even when aggregate recall is high: a small number of false negatives can represent materially different security risks from false positives.

C. LIME Text-Level Analysis

SHAP explains the downstream classifier in the 768-dimensional embedding space, whereas LIME was used to examine the original email text. For each of the four representative cases, LIME generated a local surrogate explanation by perturbing the input text and fitting a locally weighted interpretable model. The embedding model used the same maximum sequence length of 512 tokens, with batched Transformer inference during explanation generation. The analysis used 20 displayed features and a fixed random state to make the experimental procedure reproducible.

The LIME explanations provide a complementary view of the prediction because the displayed features correspond to textual tokens or token-level features rather than latent embedding dimensions. For the TP and FP cases, the highlighted text-level evidence was consistent with the strong malicious predictions, including promotional, solicitation, urgency, and call-to-action language. The TN case provided text-level evidence consistent with a legitimate business message, whereas the FN case showed that apparently suspicious textual content could nevertheless be associated with a low malicious probability when the complete contextual representation favoured the legitimate class.

LIME should nevertheless be interpreted as a local surrogate explanation rather than a causal account of the Transformer prediction. A highlighted word indicates that changing or perturbing that feature affected the local surrogate model under the sampled perturbation distribution; it does not establish that the word independently caused the original prediction. This distinction is especially important for Transformer-based models, where meaning is represented contextually rather than as an additive collection of independent word effects.

D. LIME Robustness Analysis

The robustness of the LIME explanations was assessed by repeating the analysis with 120 and 500 perturbation samples for each of the four locked test cases. The underlying hybrid model predictions remained unchanged, including the four reproduced probabilities of 0.999956, 0.000050, 0.999699, and 0.003293. Thus, the prediction itself was stable under the explanation procedure.

The individual LIME feature rankings, however, were substantially less stable. Across the four cases, the average overlap between the top-ten feature sets obtained with 120 and 500 perturbations was only 0.75 features out of 10, corresponding to a mean Jaccard similarity of approximately 0.041. For the top-20 feature sets, the average overlap was 4 features out of 20. Sign consistency was also only partial, with an average value of 0.50. The mean Pearson and

TABLE XI
LIME EXPLANATION STABILITY BETWEEN 120 AND 500
PERTURBATION SAMPLES.

Case	Top-10	Jaccard	Top-20	Jaccard	Sign
TP	1	0.0526	6	0.1765	0.5000
TN	0	0.0000	3	0.0811	0.6667
FP	2	0.1111	4	0.1111	0.5000
FN	0	0.0000	3	0.0811	0.3333

Spearman correlations of the feature-weight vectors were approximately 0.098 and 0.107, respectively.

These results indicate an important distinction between *prediction stability* and *explanation stability*. The hybrid model produced the same predictions for the locked cases, but the precise ordering of LIME features changed considerably as the perturbation sample size increased. Therefore, the LIME results should be treated as local, illustrative evidence about influential text features rather than as a stable global ranking of phishing or spam indicators.

The robustness analysis also prevents overinterpretation of individual LIME weights. Because the local surrogate depends on the perturbation distribution, sample size, and local neighbourhood, a feature’s rank can vary without implying that the underlying classifier has changed. In this study, LIME was consequently used to complement SHAP rather than to replace model-level evaluation or to establish a definitive vocabulary of malicious-email indicators.

E. Complementarity of SHAP and LIME

The two explanation methods provide different but complementary evidence. SHAP identifies which latent dimensions most strongly influence the XGBoost classifier and can therefore be used for global and local analysis of the downstream hybrid stage. LIME provides a text-facing view of individual predictions and makes the explanation easier to inspect at the level of the original email. The two methods should not be expected to produce identical feature rankings because they explain different computational representations.

The combined analysis supports a layered interpretation of the hybrid architecture. At the latent level, SHAP demonstrates that the XGBoost classifier uses a non-uniform subset of the learned DeBERTa representation. At the text level, LIME illustrates which portions of individual emails are influential within a local surrogate. The robustness experiment further shows that text-level explanations require caution: the model’s prediction can be highly stable even when the exact local feature ranking is not.

F. Summary of Explainability Findings

The explainability analysis provides three principal findings. First, SHAP identified a concentrated set of influential dimensions within the 768-dimensional DeBERTa representation, with `dim_440` having the highest mean absolute contribution. Second, local SHAP analysis demonstrated

qualitatively different attribution patterns for correct and incorrect predictions, including a security-relevant false negative and a strongly spam-like false positive. Third, LIME provided complementary text-level evidence but showed limited ranking stability when the perturbation budget was increased from 120 to 500 samples.

Overall, the findings support the use of SHAP and LIME as complementary diagnostic tools rather than as interchangeable explanation mechanisms. SHAP is better suited to analysing the behaviour of the XGBoost classifier over the learned representation, while LIME provides a more accessible local view of the source text. Neither method, on its own, establishes causal relationships between individual words, latent dimensions, and the final classification decision.

VII. DISCUSSION

The experimental findings provide a more nuanced view of Transformer–XGBoost hybridisation for email spam and phishing detection than would be obtained from accuracy alone. Across the evaluated architectures, standalone Transformers already achieved strong classification performance, while the effect of adding XGBoost depended on the underlying Transformer representation. The DeBERTa experiment further showed that hybridisation can produce a modest improvement in malicious-class detection, but the improvement should be interpreted together with the additional false positive and the computational characteristics of the modular pipeline.

A. Comparative Effect of Transformer–XGBoost Hybridisation

The comparative results show that XGBoost does not universally improve Transformer-based email classification. BERT achieved 99.4% accuracy as a standalone model, compared with 99.2% for BERT–XGBoost. Similarly, RoBERTa achieved 99.3% accuracy, whereas its hybrid achieved 99.2%. The standalone models also retained stronger balanced accuracy and MCC than their corresponding hybrids. These results indicate that a high-quality contextual representation can already contain sufficient discriminative information for the classification task, such that an additional tree-based classifier does not necessarily provide a performance advantage.

The DeBERTa results differed from the earlier BERT and RoBERTa comparisons. Standalone DeBERTa achieved 99.4% accuracy, 95.56% recall, and 96.63% F1-score, while DeBERTa–XGBoost achieved 99.5% accuracy, 97.78% recall, and 97.24% F1-score. Balanced accuracy increased from 97.67% to 98.72%, ROC-AUC increased from 99.92% to 99.98%, and PR-AUC increased from 99.38% to 99.78%. The hybrid therefore provided a modest but measurable improvement in several metrics, most importantly malicious-class recall.

The security significance of this change is clearer from the confusion matrices. Standalone DeBERTa produced two

false positives and four false negatives, whereas DeBERTa–XGBoost produced three false positives and two false negatives. Thus, the hybrid exchanged one additional false positive for two fewer false negatives. For an email-security application, this trade-off can be meaningful because false negatives correspond to malicious messages that remain undetected. Nevertheless, the preferred operating point ultimately depends on the deployment context and the relative cost assigned to false positives and false negatives.

The combined BERT, RoBERTa, and DeBERTa findings therefore support a model-dependent interpretation of hybridisation. The results do not justify the general claim that XGBoost improves every Transformer classifier. Instead, they indicate that the benefit of a downstream tree-based classifier depends on the representation produced by the Transformer and on how the classifier exploits that representation.

B. Why the DeBERTa Hybrid May Provide a Benefit

The DeBERTa–XGBoost result is consistent with the possibility that the fixed 768-dimensional Transformer representation contains nonlinear class structure that can be exploited effectively by gradient-boosted decision trees. The embedding diagnostic showed substantial variation across dimensions, with no constant or near-constant dimensions and a non-zero distribution of per-dimension standard deviations. A logistic regression diagnostic also achieved 99.5% validation accuracy and 97.18% F1-score, demonstrating that the learned representation itself was highly discriminative.

The fact that the logistic regression baseline was already strong is important. It suggests that the DeBERTa representation contains substantial linearly separable information, while the slightly different behaviour of XGBoost indicates that nonlinear partitioning of the representation can nevertheless alter the decision boundary. The improvement should not be interpreted as evidence that the Transformer representation is inadequate on its own. Rather, XGBoost provides an alternative downstream decision function over an already informative representation.

The controlled XGBoost experiments further showed that the observed improvement was not strongly dependent on small changes in tree depth or regularisation. All five evaluated configurations produced the same thresholded validation predictions, with 99.3% accuracy, 100% precision, 92.22% recall, and 95.95% F1-score. The lower-learning-rate configuration was selected because it provided the strongest ranking metrics, including ROC-AUC of 99.87% and PR-AUC of 99.06%. This supports a conservative interpretation: the final test improvement was associated with the selected downstream classifier, but the validation experiments did not reveal a large thresholded-performance difference among the tested XGBoost configurations.

C. Security-Oriented Interpretation

For security-oriented classification, aggregate accuracy can conceal the operational importance of errors in the

malicious class. The DeBERTa hybrid reduced the number of false negatives from four to two in the 1,000-message held-out test set. This corresponds to an increase in malicious-class recall from 95.56% to 97.78%. At the same time, false positives increased from two to three, reducing precision from 97.73% to 96.70%.

This trade-off is consistent with the asymmetric consequences of email-security errors. A false positive may inconvenience a legitimate user or cause a message to require additional inspection, whereas a false negative can allow a malicious message to reach a user. The results therefore suggest that the hybrid architecture may be attractive in settings where improving malicious-message recall is prioritised. However, the study does not establish an application-specific cost matrix, nor does it optimise a deployment threshold according to monetary or operational costs. The findings should consequently be interpreted as evidence of improved recall under the evaluated threshold rather than as proof of optimal security performance.

The representative error analysis also demonstrates that difficult cases are not necessarily characterised by an absence of suspicious language. The false-positive example was labelled legitimate but exhibited strongly promotional and solicitation-oriented content, while the false-negative example contained several features that appeared suspicious at the textual level. These cases reinforce the importance of contextual representation and error-level analysis rather than assuming that individual lexical cues are sufficient for reliable classification.

D. Explainability Implications

The SHAP and LIME experiments demonstrate that explainability is representation-dependent. SHAP provided a global and local view of the XGBoost stage by attributing predictions to dimensions of the 768-dimensional DeBERTa representation. This analysis identified a relatively small subset of influential latent dimensions, with `dim_440` having the largest mean absolute SHAP value. However, these dimensions are learned numerical representations and should not be assigned direct semantic labels without additional analysis.

LIME complemented the latent-space explanation by producing text-level local explanations. This makes individual predictions more accessible for human inspection, particularly when investigating false positives and false negatives. However, the robustness experiment showed that the exact LIME rankings changed substantially between 120 and 500 perturbation samples, even though the underlying predictions remained unchanged. The distinction is important: a stable prediction does not imply a uniquely stable local explanation.

Accordingly, the study supports a layered explainability strategy. SHAP is appropriate for analysing the downstream XGBoost decision process and identifying influential latent dimensions, while LIME is useful for generating illustrative text-level evidence for individual

emails. Neither method should be interpreted as a causal explanation of why the complete Transformer system produced a prediction. This is particularly relevant for security applications, where explanations may influence analyst decisions and therefore need to be presented with appropriate uncertainty.

E. Computational Implications of the Hybrid Architecture

The computational findings clarify an important distinction between representation-generation cost and downstream classifier optimisation cost. Generating the DeBERTa representations for the 5,000 training, 1,000 validation, and 1,000 test emails required approximately 264.55 minutes on the local CPU environment. In contrast, the controlled XGBoost configurations required approximately 15.9–27.9 seconds per configuration once the embeddings had been generated.

This modularity is practically useful during model development. The fixed 768-dimensional representations can be reused across multiple XGBoost configurations without repeating the expensive Transformer representation-generation stage. Consequently, researchers can perform downstream hyperparameter experiments rapidly after the embedding cache has been created. The architecture therefore provides an experimental and engineering advantage even though the Transformer remains the dominant computational component.

The results do not, however, demonstrate that the complete hybrid system has lower end-to-end inference or training cost than standalone DeBERTa. Every new email must still pass through DeBERTa before the XGBoost classifier can operate on its embedding. Furthermore, the study did not collect matched measurements of hybrid and standalone inference latency, throughput, peak memory, or energy consumption. The computational contribution should therefore be described as representation reuse and downstream optimisation efficiency rather than as proven end-to-end computational savings.

F. Relationship to the Research Gap

The findings address the identified research gap in three ways. First, the study provides a controlled comparison across BERT, RoBERTa, and DeBERTa rather than assuming that Transformer–XGBoost hybridisation is universally beneficial. Second, it evaluates explainability at both the latent representation level and the original text level using SHAP and LIME. Third, it explicitly separates the computational cost of Transformer representation generation from the cost of downstream XGBoost optimisation.

The results indicate that these three aspects are interdependent. A hybrid model may provide a useful recall improvement without reducing total computational cost, and a highly accurate model may still require careful examination of individual errors. Similarly, an explanation method may be useful diagnostically without providing stable feature rankings. The contribution of the study

therefore lies not only in the final accuracy values but also in the empirical characterisation of when and how the hybrid architecture provides value.

G. Practical Implications

For practical email-security development, the findings suggest that Transformer-XGBoost architectures should be evaluated empirically rather than adopted solely on the assumption that heterogeneous model combinations are superior. A practical workflow can first fine-tune a contextual Transformer, cache its fixed representations, and then evaluate alternative lightweight classifiers using the same representation. This allows rapid downstream experimentation while preserving the contextual modelling capability of the Transformer.

For deployments in which malicious-message recall is particularly important, the DeBERTa-XGBoost result provides evidence that the hybrid can offer a favourable operating point under the evaluated test conditions. However, deployment decisions should incorporate threshold calibration, class prevalence, analyst workload, false-positive tolerance, and the consequences of missed malicious messages. The current experiments provide a controlled research comparison rather than a production-ready operating-point analysis.

H. Overall Discussion

Taken together, the results show that Transformer-XGBoost hybridisation is neither universally beneficial nor inherently computationally cheaper. BERT and RoBERTa performed slightly better as standalone classifiers, whereas DeBERTa-XGBoost produced a modest improvement over standalone DeBERTa, particularly in malicious-class recall. The hybrid also reduced false negatives at the cost of one additional false positive.

The principal practical advantage demonstrated by the experiments is modularity. Once the computationally expensive Transformer representations have been generated, downstream XGBoost models can be trained and compared rapidly. The explainability analysis further shows that SHAP and LIME can provide complementary views of the hybrid system, provided their respective representation-level and stability limitations are recognised.

Overall, the evidence supports a conditional rather than universal argument for Transformer-XGBoost hybridisation: the architecture is most defensible when empirical evaluation demonstrates a useful downstream decision-boundary improvement, when representation reuse is valuable during model development, and when explainability requirements justify analysing both latent and text-level evidence.

VIII. LIMITATIONS AND FUTURE WORK

The findings of this study should be interpreted within the scope of the experimental design and available computational evaluation. Several limitations affect the generalisability, statistical strength, computational interpretation,

and explainability claims of the results. These limitations also define several direct directions for future research.

A. Limitations

1) *Controlled Experimental Dataset*: The Transformer experiments were conducted using a controlled 7,000-email subset comprising 5,000 training messages, 1,000 validation messages, and 1,000 held-out test messages, rather than training and evaluating the Transformer models on the full integrated corpus of 258,656 unique messages. Consequently, the reported Transformer results demonstrate performance under the controlled experimental setting but do not establish performance at full-corpus scale.

In addition, the integrated datasets are principally established spam/ham benchmarks. Although the malicious class is used throughout the study for the security-oriented binary classification task, it should not be interpreted as a comprehensive representation of contemporary phishing attacks. Modern phishing campaigns can contain characteristics that are not fully represented by conventional benchmark corpora.

2) *Statistical Validation*: The study did not conduct statistical significance testing, confidence intervals, or repeated independent train/test splits to determine whether the relatively small differences between some of the evaluated models are statistically robust. This is particularly relevant to the BERT, RoBERTa, and DeBERTa comparisons, where several performance differences are small. The results should therefore be interpreted as empirical outcomes of the controlled experimental protocol rather than as evidence of statistically significant superiority across repeated samples.

3) *Computational Benchmarking*: The computational analysis measured Transformer representation extraction and downstream XGBoost training, with approximately 264.55 minutes required to generate the DeBERTa training, validation, and test representations in the local CPU environment and approximately 15.9–27.9 seconds required for the evaluated XGBoost configurations. However, the study did not conduct a matched end-to-end benchmark comparing standalone DeBERTa with the complete DeBERTa-XGBoost pipeline.

In particular, equivalent measurements of inference latency per email, throughput, peak memory consumption, CPU/GPU utilisation, and energy consumption were not collected for both architectures. Therefore, the study does not establish that the hybrid architecture reduces total training or inference cost. Its demonstrated computational advantage is limited to representation reuse and rapid downstream classifier optimisation after the Transformer representations have been generated.

4) *Explainability Stability*: The LIME robustness analysis compared explanations generated using 120 and 500 perturbation samples for four representative test cases. Although the underlying predictions remained stable, the exact feature rankings showed limited stability. The relatively small number of cases and perturbation settings

limits the extent to which these findings can be generalised to all messages.

Accordingly, individual LIME feature rankings should be interpreted cautiously. The results support the use of LIME as a local and illustrative diagnostic method, but they do not establish a stable global ranking of textual indicators for phishing or spam. Similarly, the SHAP dimensions identified in the latent representation should not be assigned direct semantic meanings without additional representation-level analysis.

5) *Security and Deployment Scope*: The study did not evaluate several factors that are important for deployment-scale cybersecurity systems. These include adversarial robustness, concept drift, multilingual email, attachment-based threats, URL reputation, sender authentication signals, and deployment-scale throughput. The present results therefore represent a controlled classification study rather than a complete enterprise email-security evaluation.

B. Future Work

1) *Additional Transformer Architectures*: Future work should extend the comparative evaluation to additional Transformer architectures and computationally efficient representation models. In particular, future experiments should investigate whether models with lower representation-generation costs can preserve the semantic and classification advantages observed with DeBERTa while making the overall system more practical for resource-constrained environments.

This extension would also allow the model-dependent nature of Transformer-XGBoost hybridisation to be investigated more systematically. Rather than assuming that a particular downstream classifier is universally advantageous, future studies can evaluate whether the observed hybrid benefit varies according to model architecture, representation dimensionality, fine-tuning strategy, and classifier design.

2) *Matched End-to-End Computational Evaluation*: A major priority for future work is a controlled end-to-end computational benchmark. Such an evaluation should compare standalone Transformer classification with Transformer-XGBoost classification under the same hardware and software conditions. Measurements should include latency per email, emails processed per second, peak memory consumption, CPU/GPU utilisation, and energy consumption.

These measurements would directly test whether replacing the native Transformer classification stage with XGBoost provides a meaningful operational advantage. They would also distinguish the demonstrated downstream optimisation benefit from any potential end-to-end deployment benefit.

3) *Statistical Confidence and Repeated Evaluation*: Future experiments should use repeated stratified train/test splits, repeated trials, or cross-validation where appropriate to quantify variability in model performance. Confidence

intervals and statistical significance testing could then be used to determine whether small differences between standalone and hybrid architectures are robust.

This would be particularly valuable for interpreting cases where the difference in accuracy or F1-score is only a fraction of a percentage point. A repeated evaluation framework would also reduce dependence on the particular random split used in the current controlled benchmark.

4) *Dedicated Phishing and Temporal Evaluation*: Future research should expand the evaluation beyond general spam/ham corpora to dedicated phishing datasets containing contemporary malicious-email campaigns. Temporal evaluation should also be considered so that models trained on earlier messages can be tested on later messages. Such an evaluation would provide evidence about concept drift and the ability of contextual representations to generalise across changing attack patterns.

Adversarial evaluation should further examine whether malicious actors can deliberately modify textual characteristics to evade the Transformer and hybrid classifiers. This would provide a stronger assessment of robustness in realistic security environments.

5) *Extended Explainability Evaluation*: Further work should compare explainability across standalone Transformer classifiers and their XGBoost hybrids. Such experiments could investigate whether replacing the native classification head changes the latent dimensions and textual evidence associated with individual predictions.

Future explainability studies should also examine explanation stability across larger numbers of test cases and multiple perturbation settings. The objective should be to determine whether explanations remain useful under realistic variation in the input distribution and whether explanations can support security analysts without being interpreted as causal ground truth.

C. Summary of Limitations and Future Work

The present study provides controlled evidence concerning the performance, computational modularity, and explainability of Transformer-XGBoost architectures for email security, but several boundaries remain. The controlled 7,000-message Transformer benchmark, absence of repeated statistical validation, lack of matched end-to-end computational measurements, limited LIME robustness evaluation, and restricted security deployment scope constrain the strength of the conclusions.

Future research should therefore move from controlled comparative evaluation toward broader and more deployment-oriented validation. The most important priorities are computationally efficient Transformer representations, matched end-to-end benchmarking, repeated statistical evaluation, dedicated and temporally separated phishing datasets, adversarial testing, and more extensive explainability robustness analysis. These extensions would provide stronger evidence about whether the modest DeBERTa-XGBoost improvement observed in this study

can translate into reliable operational benefits in real-world email-security systems.

IX. CONCLUSION

This study presented a comparative investigation of BERT, RoBERTa, and DeBERTa-v3-base, together with corresponding XGBoost-based hybrid variants, for binary email spam and malicious-email detection. The study was motivated by the need to determine whether combining contextual Transformer representations with a separate tree-based classifier provides measurable benefits in classification performance, computational workflow, and explainability. Rather than assuming that hybridisation is universally advantageous, the study evaluated the architectures comparatively under a controlled experimental protocol.

The results demonstrate that the effect of XGBoost depends on the underlying Transformer representation. In the BERT and RoBERTa experiments, the standalone Transformers achieved higher principal classification performance than their corresponding hybrid variants. BERT achieved 99.40% accuracy compared with 99.20% for BERT-XGBoost, while RoBERTa achieved 99.30% compared with 99.20% for RoBERTa-XGBoost. These findings show that adding a downstream tree-based classifier does not automatically improve an already strong contextual representation.

The DeBERTa experiment produced a more favourable hybrid outcome. Standalone DeBERTa achieved 99.40% accuracy, 95.56% malicious-class recall, 96.63% F1-score, and 97.67% balanced accuracy. DeBERTa-XGBoost achieved 99.50% accuracy, 97.78% recall, 97.24% F1-score, and 98.72% balanced accuracy. The hybrid also improved ROC-AUC from 99.92% to 99.98% and PR-AUC from 99.38% to 99.78%. Most importantly from a security perspective, the number of false negatives decreased from four to two on the held-out test set, although false positives increased from two to three. The evidence therefore supports a modest security-oriented improvement rather than a universal performance advantage.

The computational analysis provides an equally important qualification. Generating the DeBERTa representations for the controlled dataset required approximately 264.55 minutes on the local CPU environment, demonstrating that the Transformer remained the dominant computational component. Once the 768-dimensional representations had been generated, however, the five controlled XGBoost configurations required only approximately 15.9–27.9 seconds each. The practical computational benefit of the hybrid architecture is consequently representation reuse and inexpensive downstream classifier optimisation. The study did not demonstrate lower end-to-end training or inference cost because matched measurements of latency, throughput, memory consumption, and energy were not collected.

The explainability analysis further demonstrated the value and limitations of complementary explanation tech-

niques. SHAP identified the latent DeBERTa dimensions most influential to the XGBoost classifier, with `dim_440` having the largest mean absolute SHAP value. These dimensions represent learned latent numerical features and cannot be interpreted directly as individual semantic concepts. LIME provided a text-level view of individual predictions and was particularly useful for examining representative true-positive, true-negative, false-positive, and false-negative cases. However, the robustness analysis showed that the precise LIME feature rankings changed substantially when the perturbation budget increased from 120 to 500 samples, despite the underlying model predictions remaining stable. LIME should therefore be treated as local and illustrative evidence rather than as a definitive global ranking or causal explanation.

Overall, the study answers the central research question conditionally. Transformer-XGBoost hybridisation can provide useful benefits, but those benefits are dependent on the Transformer representation and classifier configuration. The strongest evidence in the present experiments comes from DeBERTa-XGBoost, which produced a modest improvement in malicious-class recall and reduced false negatives relative to standalone DeBERTa. At the same time, the results do not support the claim that XGBoost universally improves Transformer classification or that it reduces the complete computational cost of the detection system.

The principal contribution of the study is therefore a controlled empirical characterisation of the conditions under which Transformer-XGBoost hybridisation is useful for email security. The findings support a modular architecture in which a contextual Transformer generates reusable representations and XGBoost provides a lightweight downstream decision layer. Future work should extend this evaluation to larger and dedicated phishing datasets, repeated statistical validation, temporal and adversarial testing, and matched end-to-end computational benchmarks. Such studies would establish whether the modest improvements observed in the controlled DeBERTa experiment can translate into reliable operational benefits in real-world email-security environments.

ACKNOWLEDGMENT

The authors acknowledge the University of the West of Scotland and the project supervision and academic support provided during the development of this research.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- [2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.

- [3] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with disentangled attention,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [4] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [5] S. Gualberto, R. T. Sousa Júnior, T. P. B. Vieira, J. P. J. Costa, and C. Duque, “From feature engineering and topic models to enhanced prediction rates in phishing detection,” *IEEE Access*, 2020.
- [6] P. Bountakas, K. Koutroumpouchos, and C. Xenakis, “A comparison of natural language processing and machine learning methods for phishing email detection,” 2021.
- [7] P. L. Indrasiri, M. Halgamuge, and A. Mohammad, “Detection of e-mail phishing attacks using machine learning and deep learning,” 2022.
- [8] I. AbdulNabi and Q. M. Yaseen, “Spam email detection using deep learning techniques,” 2021.
- [9] S. Bagui, D. Nandi, S. Bagui, and R. J. White, “Machine learning and deep learning for phishing email classification using one-hot encoding,” 2021.
- [10] D. Rathee and S. Mann, “An analysis of machine learning algorithms and deep neural networks for email spam classification using natural language processing,” 2022.
- [11] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, “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*, 2016, pp. 1135–1144.
- [13] Carnegie Mellon University, “Enron Email Dataset,” CALO Project corpus. [Online]. Available: <https://www.cs.cmu.edu/~enron/>
- [14] Apache SpamAssassin Project, “SpamAssassin Public Corpus.” [Online]. Available: <https://spamassassin.apache.org/old/publiccorpus/>
- [15] Conference on Email and Anti-Spam, “CEAS 2008 Spam Filtering Challenge,” 2008. [Online]. Available: <https://www.ceas.cc/2008/>