Ingénierie des Systèmes d'Information
Vol. 31, No. 3, March, 2026, pp. 901-910
Journal homepage: http://iieta.org/journals/isi

DistilBERT for Efficient and Accurate Email Phishing Detection: A Benchmark Against Machine and Deep Learning Models

Dam Minh Linh[1], Han Minh Chau[2*], Le Ha Thanh[1], Nguyen Thi Bich Nguyen[1], Lam Duy Quy[1]

[1] Information Security Technology Lab and Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Ho Chi Minh City 700000, Vietnam
[2] Faculty of Information Technology, HUTECH University, Ho Chi Minh City 700000, Vietnam

Corresponding Author Email: hm.chau80@hutech.edu.vn

https://doi.org/10.18280/isi.310321

Received: 3 December 2025
Revised: 10 February 2026
Accepted: 19 March 2026
Available online: 31 March 2026

Keywords:
phishing email detection, DistilBERT, lightweight transformer, comparative benchmarking, computational efficiency, real-time cybersecurity

ABSTRACT

Email phishing remains a persistent cybersecurity threat that exploits human vulnerabilities, often evading technical safeguards. While machine learning (ML) and deep learning (DL) have been widely applied for phishing detection, systematic benchmarks comparing lightweight transformer models with traditional approaches remain limited. This study addresses this gap by evaluating six models—Naïve Bayes, Random Forest, XGBoost, LSTM, BiLSTM, and a fine-tuned DistilBERT—on a real-world dataset of 17,538 emails using three train-test splits (60:40, 70:30, 80:20). DistilBERT consistently outperforms all baselines across all splits. Under the 80:20 split, it achieves the highest accuracy (98.77%), precision (99.10%), recall (98.97%), F1-score (99.02%), and AUC (99.91%). Remarkably, it maintains low computational overhead with a training time of 342 seconds, demonstrating an optimal trade-off between detection accuracy and efficiency. In contrast, BiLSTM, the best-performing recurrent model, reaches 97.43% accuracy but produces more false negatives—a more critical security risk than false positives in phishing detection. Additional experiments reveal that DistilBERT maintains stable performance across different data splits, with AUC values consistently above 0.998. The confusion matrix analysis shows that DistilBERT misclassifies only 25 legitimate emails as phishing (false positives) and misses only 23 phishing emails (false negatives), significantly outperforming all baseline models. These findings demonstrate that lightweight transformer models like DistilBERT offer a practical, scalable, and cost-effective solution for real-time phishing email detection, effectively bridging the gap between high accuracy and production-ready deployability.

1. INTRODUCTION

Email phishing remains a widespread cyber threat, driven by social engineering and online deception to compromise sensitive information. According to the Phishing Activity Trends Report for Q4 2024 by the Anti-Phishing Working Group, the number of unique phishing email campaigns reached 28,327 in October, 27,668 in November, and increased to 33,899 in December, suggesting a rising trend toward the end of the year [1].

The authors [2] reports that phishing and pretexting (BEC) accounted for approximately 73% of social engineering–related breaches, with phishing alone responsible for 31%. Alarmingly, users typically fall for phishing emails in under 60 seconds—21 seconds to click a malicious link and an additional 28 seconds to submit their data. Furthermore, about 20% of users reported phishing attempts in security simulations, and among those who clicked the phishing email, 11% still reported it. Financially motivated incidents involving BEC accounted for around 24–25% of such attacks, with median transaction values of approximately USD 50,000.

Machine learning–based cybersecurity systems may exhibit vulnerabilities under real-world conditions, posing challenges for reliable phishing detection [3]. Various learning-based approaches have been explored for phishing email detection in practical settings [4]. However, traditional methods are often limited by handcrafted features and relatively small datasets [5]. A Naïve Bayes–based method has been reported to achieve up to 97% accuracy (Acc) [6].

The findings [7] further demonstrated that deep learning (DL) with natural language processing can reach 98.2% Acc in phishing email detection. Similarly, the method in study [8] developed an efficient filtering framework for spam and phishing emails using DL. However, detecting phishing at the individual message level requires semantic text analysis, where recent transformer-based models show the greatest promise.

The Meta Phishing Detector Agent, developed on the MetaGPT framework, leverages large language models (LLMs) as the core agent for phishing email detection [9]. Studies employing BERT and LSTM have demonstrated the effectiveness of DL in enhancing phishing detection [10], while LLM-generated emails introduce new security concerns [11]. Recently, the APOLLO framework achieved 97% Acc

for phishing email detection. This trend underscores the potential of LLM-based approaches, aligning with recent findings [12], and complements our work, where we propose a fine-tuned DistilBERT model to improve the Acc and reliability of phishing email detection.

In a practical setting, the proposed model can be used as a classification component within an email filtering pipeline, where incoming emails are preprocessed and classified as phishing or legitimate. This component can be integrated into a mail gateway or a server-side filtering service for real-time detection.

While phishing detection has been widely studied, there remain several limitations. First, few works provide a systematic comparison across machine learning (ML), DL, and Transformer-based approaches for email phishing detection. Second, many prior studies rely on relatively small datasets, limiting the generalizability of their results. Third, lightweight Transformer models such as DistilBERT have received limited attention despite their potential for practical deployment. To overcome these limitations, this study defines three main objectives. First, it adapts a DistilBERT model for phishing email classification. Second, it evaluates the proposed approach against representative machine learning and deep learning methods, including NB, RF, XGB, LSTM, and BiLSTM. Third, it performs a detailed performance assessment using multiple evaluation metrics, including Acc, Precision (Pre), Recall (Rec), F1-score, Loss, ROC curves, and confusion matrices (CM). The main contribution of this study is the development of a consistent benchmarking framework that enables a fair comparison across ML, DL, and transformer-based approaches, while providing empirical evidence of the advantages of lightweight Transformers like DistilBERT for robust and scalable phishing email detection.

The structure of the paper is outlined as follows. Section 2 reviews existing studies, including ML, DL, and transformer-based approaches. Section 3 describes the proposed method, covering the experimental workflow, evaluation metrics, dataset, and the fine-tuned DistilBERT architecture. Section 4 presents the performance evaluation and discusses the experimental results, including comparisons with prior work. Section 5 concludes the study. The paper also includes an acknowledgment section followed by the references.

2. LITERATURE REVIEW

Findings in the study [13] emphasized that while email remains one of the most essential channels for digital communication, its openness also makes it susceptible to misuse through spam and phishing attacks. Despite numerous studies proposing diverse approaches—from metadata analysis to content-based detection—the lack of consistency in datasets, terminology, and feature representations hinders meaningful comparison and reproducibility. Building on this context, research on phishing email detection has evolved from classical ML methods to modern DL and transformer-based architectures. This section reviews these approaches, outlining their respective strengths, limitations, and relevance to the present study.

2.1 Machine learning approaches

A study [14] employed a dataset consisting of 501 phishing and 4,090 legitimate emails, applying TF-IDF, Word2Vec, and Doc2Vec features in combination with ML models such as Support Vector Machine (SVM), RF, and XGB. Similarly, the work [15] proposed a multilingual spam and phishing detection system using NB in two stages: (i) language identification (Arabic, English, and Chinese) and (ii) phishing classification. Evaluated on a dataset of 2,000 emails, their model achieved 98.4% Acc, with a 0.08% false positive rate and 2.90% false negative rate. In another study, Jáñez-Martino et al. [16] introduced two spam email datasets of 15,000 messages each in English and Spanish, demonstrating that Term Frequency–Inverse Document Frequency (TF-IDF) combined with Logistic Regression (LR) or NB achieved Acc up to 98.5%. More recently, a study [17] conducted a study on phishing email detection in the Bangla language using a dataset of 5,572 emails, comprising 4,572 legitimate (ham) and 1,000 phishing emails. DL models in this study outperformed traditional ML, with BiLSTM achieving the highest Acc of 97% and an F1-score of 88.89%, followed by Convolutional Neural Network (CNN) at 96.8% Acc (F1-score = 87.45%) and Bangla-BERT at 96.4% Acc (F1-score = 86.35%). In contrast, conventional ML models such as NB, K-Nearest Neighbors (*k*-NN), Decision Tree (DT), SVM, AdaBoost, and RF achieved accuracies below 93.6%.

Although these ML approaches achieved encouraging Acc across different datasets and languages, they are constrained by several limitations. Most studies rely on relatively small or language-specific datasets, which restrict the generalization of results to large-scale, real-world email traffic. Moreover, ML methods depend heavily on handcrafted feature extraction (e.g., TF-IDF, Word2Vec), which often fails to capture deeper semantic and contextual patterns in phishing emails. These gaps underscore the need for more advanced methods, such as DL and Transformer-based architectures, that can automatically learn richer linguistic representations.

2.2 Deep learning models

The study [18] evaluated LSTM and BiLSTM models on a balanced dataset of 4,000 emails from the Enron and Monkey.org corpora (2,000 ham and 2,000 phishing). BiLSTM outperformed LSTM, reaching an Acc of 98.35% and an F1-score of 98.24%, whereas the LSTM model achieved lower results, with Acc below 97% and F1-score under 96%. Building on this trend, Remmide et al. [19] further advanced phishing detection by combining BiLSTM with a Temporal Convolutional Network (TCN), reaching an Acc of 98.28%. Similarly, in a related effort, McGinley and Monroy [20] applied a CNN to datasets including Enron-Spam, Enron, and Nazario (3,804 emails: 1,870 ham and 1,934 phishing), achieving an Acc of 98.14% and an F1-score of 98.16%.

Further experimentation [21] conducted experiments on a large-scale Enron dataset of 33,727 emails (16,563 ham and 17,188 phishing), where a hybrid DNN–BiLSTM achieved an Acc of 98.69% and an F1-score of 98.69%, substantially outperforming classical models such as LR, RF, RNN, CNN, and LSTM, which all recorded accuracies below 96.39%. Extending this line of research, Borra et al. [22] extended the evaluation across Enron, CLAIR, and Hate Speech & Offensive datasets (32,427 emails) by comparing CNN with traditional classifiers (LR, SVM, NB, AdaBoost). Their CNN model achieved an Acc of 98.43% and an F1-score of 97.07%, whereas the traditional ML models underperformed with accuracies below 89% and an F1-score of only 80.66%.

DL approaches achieve strong performance with accuracies

above 98%, confirming the effectiveness of CNN, BiLSTM, and hybrid DNN for phishing detection. However, they remain sensitive to dataset size and imbalance, and sequential models like LSTM and BiLSTM incur high computational costs. While CNN and hybrid methods offer efficiency, they struggle to capture complex contextual dependencies. These limitations motivate the use of Transformers, which exploit attention mechanisms for better scalability and generalization.

2.3 Transformer-based architectures

The work of Sanh et al. [23] proposed DistilBERT, a compressed version of BERT that reduces model size by 40% while preserving 97% of its language understanding capabilities and providing 60% faster inference. Unlike previous approaches that mainly applied distillation to task-specific models, their framework introduced a knowledge transfer strategy during the pre-training stage, integrating language modeling, representation alignment, and cosine-similarity objectives.

Building on the success of Transformer-based models, Mehdi Gholampour and Verma [24] evaluated BERT and its variants (ALBERT, RoBERTa, DeBERTa, DistilBERT) on the IWSPA 2.0 and a generated dataset comprising 7,286 emails, achieving 98–98.8% Acc and F1Ss between 92% and 97%. Similarly, in a related investigation, AbdulNabi and Yaseen [15] applied BERT, BiLSTM, *k*-NN, and NB to the UCI ML and SpamFilter datasets (5,000 emails), where BERT attained 97.3% Acc and an F1-score of 96.96%, outperforming BiLSTM (96.43%, F1-score = 0.96) and classical ML models (*k*-NN and NB with < 94% Acc and F1-score < 0.94).

Motivated by these findings, our study fine-tunes DistilBERT specifically for phishing email detection and benchmarks it against ML and DL baselines. The results demonstrate that Fine-tuned DistilBERT not only maintains the efficiency of lightweight Transformer architectures but also achieves state-of-the-art performance across Acc, Pre, Rec, and F1-score, underscoring its suitability for robust and scalable phishing detection.

3. PROPOSED METHOD

3.1 Experimental workflow

Figure 1 illustrates the experimental workflow for email phishing detection, which integrates both traditional ML and DL approaches in a unified process. The workflow begins with the input email dataset, followed by a preprocessing phase involving the removal of duplicate and null entries (resulting in 17,538 valid samples), binary label encoding, and text normalization through hyperlink, punctuation, and whitespace removal, lowercase conversion, and stop-word filtering. In this stage, tokenization is automatically handled by the *DistilBertTokenizerFast* module during model preprocessing, and the cleaned dataset is saved in UTF-8 format to ensure reproducibility across experiments.

In the ML branch, textual features are extracted using TF–IDF vectorization with configurations such as unigrams/bigrams, document frequency thresholds, and stopword removal, while in the DL branch, emails are tokenized using the DistilBERT tokenizer and subjected to padding or truncation to a specified maximum length (MAX_LEN). The processed data is then stratified into training and testing sets with ratios of 60:40, 70:30, and 80:20 for comparative evaluation. Subsequently, ML models (RF, NB, XGB) and DL models (LSTM, BiLSTM, Fine-tuned DistilBERT) are trained and assessed using performance metrics including Acc, Pre, Rec, F1-score, Confusion Matrix, and ROC curve, enabling a comprehensive comparison to determine the best-performing model across different splits.

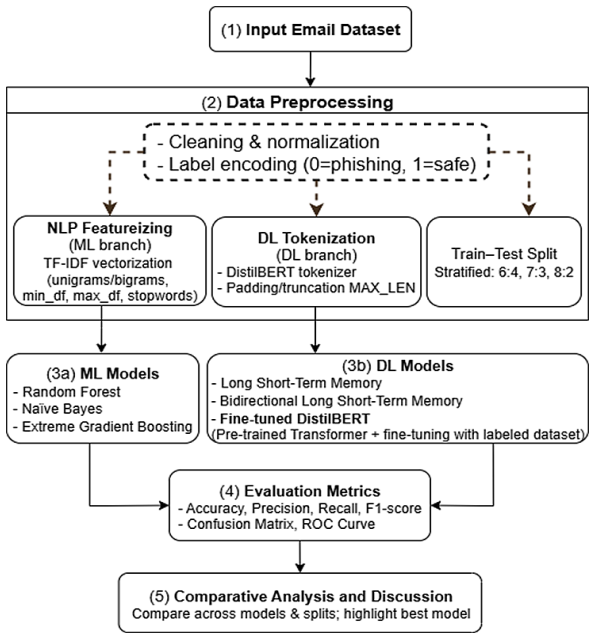

Figure 1. Experimental workflow for email phishing detection integrating machine learning (ML) and deep learning (DL) approaches

3.2 Evaluation metrics

Model performance is assessed using confusion-matrix–based measures (TP, TN, FP, FN). Acc quantifies overall correctness. Pre reflects how many predicted phishing emails are truly phishing, whereas Rec (sensitivity) captures the proportion of phishing emails correctly identified. The F1-score summarizes the trade-off between Pre and Rec. Additionally, the Area Under the ROC Curve (AUC) is reported to evaluate class separability across decision thresholds.

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision\ (Pre) = \frac{TP}{TP + FP} \tag{2}$$

$$Recall\ (Rec) = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score\ (F1S) = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

$$AUC = \int_0^1 TPR(f)\ df \tag{5}$$

This evaluation framework is consistent with established methodologies in phishing detection and cybersecurity classification, thereby reinforcing the reliability and validity of the employed performance indicators [25, 26].

3.3 Dataset

This study utilized a publicly available phishing email dataset released by Kirollos Ashraf on Kaggle [27]. The dataset contains 17,538 email samples, consisting of 10,980 legitimate (safe) emails and 6,558 phishing emails. Each email instance was pre-labeled as either phishing or benign, enabling supervised ML tasks. The dataset reflects diverse phishing strategies, such as deceptive subject lines, malicious links, and fraudulent sender information, thereby offering a realistic distribution of phishing attempts. Its composition ensures a meaningful comparison across models while still preserving the natural imbalance commonly observed in real-world phishing detection scenarios.

Table 1 presents the class-wise distribution of benign and phishing emails across three train–test ratios. For the 60:40 split, the training set comprises 10,523 samples (6,588 benign and 3,935 phishing), and the test set contains 7,015 samples (4,392 benign and 2,623 phishing). Increasing the training proportion to 70:30 produces 12,277 training and 5,261 testing samples, whereas the 80:20 split yields the largest training set (14,030 emails) and the smallest test set (3,508 emails). This stratified sampling strategy ensures balanced representation of both classes across all splits, supporting consistent and reliable model evaluation.

Table 1. Class-wise sample distribution for safe and phishing emails across various train–test ratios

Train–Test Ratio	Safe Email (Train)	Phishing Emails (Train)	Total (Train)	Safe Emails (Test)	Phishing Email (Test)	Total (Test)
60:40	6,588	3,935	10,523	4,392	2,623	7,015
70:30	7,686	4,591	12,277	3,294	1,967	5,261
80:20	8,784	5,246	14,030	2,196	1,312	3,508

Table 2. Model configuration and hyperparameter settings

Attribute	Naïve Bayes	Random Forest	XGBoost	LSTM	BiLSTM	Fine-tuned DistilBERT
Input representation	TF-IDF	TF-IDF	TF-IDF	Tokenized	Tokenized	Tokenized (DistilBERT tokenizer)
Max features / vocab	10,000	10,000	10,000	Vocab size	Vocab size	Pretrained vocab
Max sequence length	–	–	–	150	150	256
Embedding dimension	–	–	–	50	50	768 (pretrained)
Model structure	NB	RF	XGB	LSTM	BiLSTM	DistilBERT
Hidden size	–	–	–	100	100	768
Dropout	–	–	–	0.5	0.5	–
Output layer	Probabilistic	Ensemble	Boosted	Sigmoid	Sigmoid	Linear + Softmax
Loss function	Log loss (cross-entropy)	Log loss (cross-entropy)	Log loss (cross-entropy)	BCE	BCE	Cross-entropy
Optimizer	Default settings	Default settings	Default settings	Adam	Adam	AdamW
Learning rate	Default settings	Default settings	Default settings	Default Adam (0.001)	Default Adam (0.001)	3×10^{-5}
Weight decay	–	–	–	–	–	0.01
Batch size	–	–	–	–	–	32
Epochs	–	–	–	10	10	3
Train: test splits	60:40 / 70:30 / 80:20	same	same	same	same	same

Note: NB, RF, and XGB were implemented using default library settings. For LSTM and BiLSTM, the configurations correspond to the final models used in the experiments. DistilBERT was fine-tuned using the HuggingFace Transformers framework.

3.4 Fine-tuned DistilBERT architecture

The configuration in Table 2 includes model settings, training hyperparameters, and reproducibility controls such as random seed and train–test splits, ensuring stable optimization, reliable evaluation, and consistency across experiments.

The proposed algorithm integrates preprocessing, tokenization, training, and evaluation into a unified pipeline for phishing email detection. The Fine-tuned DistilBERT model, optimized with GPU acceleration, mixed Pre, and gradient accumulation, demonstrates both high effectiveness and stable performance across multiple train–test ratios, as summarized in Algorithm 1.

LSTM and BiLSTM were selected for their ability to capture long-term sequential dependencies and bidirectional contextual relationships within email text. The transformer-based model DistilBERT- comprising six transformer layers, twelve attention heads, a hidden size of 768, and approximately 66 million parameters was prioritized over larger variants such as BERT-base and RoBERTa due to its compact six-layer architecture, which reduces the parameter count by about 40% while preserving over 95% of BERT's performance. This balance between Acc and computational efficiency makes DistilBERT particularly suitable for real-time and scalable phishing email detection systems.

Table 2 summarizes the configuration settings of all evaluated models in a unified manner. Traditional machine learning models (NB, RF, and XGB) were trained on TF-IDF features with a maximum of 10,000 dimensions. LSTM and

BiLSTM models were trained on tokenized sequences with a maximum length of 150 and an embedding dimension of 50. The DistilBERT model was fine-tuned using a learning rate of 3×10^{-5}, batch size of 32, 2 training epochs, and a maximum sequence length of 256.

Algorithm 1 summarizes the DistilBERT training and inference pipeline to ensure clarity and reproducibility.

Algorithm 1: DistilBERT Training and Inference Pipeline for Email Phishing Detection

Input:

- Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the email text ("Email Text") and $y_i \in \{0,1\}$ is the class label ("Email Type"; 0 = phishing, 1 = safe).
- Configuration parameters: MODEL_NAME = distilbert-base-uncased, MAX_LEN, BATCH_SIZE , NUM_EPOCHS , LR, WEIGHT_DECAY, WARMUP_RATIO, MAX_GRAD_NORM, SEED, GRAD_ACCUM_STEPS.
- Train-test ratios $\mathcal{S} = \{0.6, 0.7, 0.8\}$ (corresponding to test ratios $0.4, 0.3, 0.2$).

Output:

For each split $\forall s \in \mathcal{S}$:

- Best-performing model $\mathcal{M}^*$ with saved parameters and tokenizer.
- Performance metrics on the test set: Loss, Acc, Pre, Rec, F1-score, ROC–AUC; confusion matrix, ROC curve plots, and CSV summary.

Step 1 – Initialization and Reproducibility

- Fix seeds (set_seed(SEED)), numpy, torch, detect GPU availability, and enable cudnn.benchmark.
- Configure automatic mixed Pre (AMP):

amp_dtype = $\begin{cases} \text{bfloat16, if Compute Capability} \geq 8.0 \\ \text{float16, otherwise} \end{cases}$

- Enable GradScaler when using fp16.

Step 2 – Label Normalization and Extraction

2.1: Map all textual labels to binary form $\{0, 1\}$.

2.2: Extract $X \leftarrow$ list of email texts, $y \leftarrow$ list of binary labels.

Step 3 – Tokenization and Preprocessing

3.1: Load the tokenizer $\mathcal{T} \leftarrow$ AutoTokenizer(MODEL_NAME, use_fast = True).

3.2: Define BatchedTokenize to process X in batches, truncating to MAX_LEN and returning input_ids and attention_mask.

Step 4 – Loop Over Train – Test Splits

for $s \in \mathcal{S}$ **do**:

4.1: Define split tag $t = \text{f"}\{int(round(10 * s))\}\_\{int(round(10 * (1 - s)))\}"$ (e.g., $60{:}40, 70{:}30, 80{:}20$), create directories for models, figures, and CSV files.

4.2: Perform a stratified split: $(X^{tr}, X^{te}, y^{tr}, y^{te}) \leftarrow$ TrainTestSplit(X, y; train_size $= s$, stratify $= y$, random_state = SEED)

4.3: Pre-tokenize: $(I^{tr}, A^{tr}) \leftarrow \text{BatchedTokenize}(X^{tr})$, $(I^{te}, A^{te}) \leftarrow \text{BatchedTokenize}(X^{te})$.

4.4: Construct PyTorch Dataset and Dataloader for training and testing, enabling GPU optimizations (pin_memory, prefetch_factor, persistent_workers).

4.5: Initialize DistilBERT $\mathcal{M}$ with num_labels = 2 and load to device . Optionally compile with torch.compile when available.

4.6: Create AdamW optimizer and linear scheduler with warm-up (WARMUP_RATIO).

4.7: Training loop:

for epoch $e = 1 \rightarrow$ NUM_EPOCHS **do**

(a) Set $\mathcal{M}$ to training mode; initialize running_loss **and** seen**; zero optimizer gradients.**

(b) for each batch $\mathcal{B}$ in train loader **do**

- Move $\mathcal{B}$ to device.
- **AMP region:** Compute $l \leftarrow \frac{\text{loss}(\mathcal{M}(\mathcal{B}))}{\text{GRAD\_ACCUM\_STEPS}}$
- If bf16: backprop directly; else scale loss and backprop with GradScaler.
- Every GRAD_ACCUM_STEPS steps: unscale (if fp16), clip gradients (MAX_GRAD_NORM), update weights, zero gradients, and step the scheduler.
- Update running_loss and seen.

(c) **Validation:** Evaluate $\mathcal{M}$ on the test loader to obtain $F1_{\text{val}}$. If $F1_{\text{val}}$ improves, save $\mathcal{M}$, tokenizer, and best metrics.

end for

4.8: **Final evaluation:** Run full inference on the test set to compute Loss, Acc, Pre, Rec, F1-score, ROC–AUC; output classification report and confusion matrix.

4.9: **Visualization and saving:** Plot and save ROC curve and confusion matrix; export CSV summary with results for split s.

end for

Complexity Analysis

- Preprocessing and tokenization: $O(N \cdot L)$, where $L =$ MAX_LEN.
- Training: Let $E =$ NUM_EPOCHS, B = BATCH_SIZE, and d be the hidden size of DistilBERT. The per-epoch step count is $\approx \lceil N_{tr}/B \rceil$.
- Each forward-backward pass costs $O(L \cdot d^2 + L^2 \cdot d)$ due to self-attention.

Total training cost:

$T_{\text{train}} = O\left(|\mathcal{S}| \cdot E \cdot \frac{N_{tr}}{B} \cdot \text{Cost}_{\text{forward+backward}}(L, d)\right)$.

- Inference: $O\left(N_{te} \cdot \text{Cost}_{\text{forward}}(L, d)\right)$.

Memory: Proportional to $B \cdot L \cdot d$ plus model parameters.

4. PERFORMANCE EVALUATION AND DISCUSSION

The experiments were conducted on a workstation featuring dual Intel Xeon E5-2696 v3 CPUs (2.30 GHz, 36 cores, 72 threads) with 64 GB DDR4 RAM. To maintain system stability under heavy workloads, an ASUS TUF 1200 W power supply was used. Computational acceleration for training and inference was provided by an NVIDIA GeForce RTX 3090 XC3 Ultra GPU (24 GB GDDR6X, 10,496 CUDA cores).

4.1 Experimental results across machine learning, deep learning, and fine-tuned DistilBERT

The comparison in this study reflects differences across

complete processing pipelines, including feature representation and model architecture, rather than a strictly controlled comparison of classifier models. Traditional models use TF-IDF features, whereas the DistilBERT model relies on contextual embeddings, and LSTM-based models use their own embedding configurations. Therefore, the results should be interpreted as a pipeline-level comparison.

Table 3 presents a comparative analysis of model performance across multiple training–testing ratios. The fine-tuned DistilBERT consistently achieves the highest Acc, recording 98.77%, 98.71%, and 98.63% for the 60:40, 70:30, and 80:20 splits, respectively. For the 60:40 configuration, BiLSTM attains 97.38%, outperforming LSTM (95.64%) and surpassing the classical models RF, NB, and XGB, which yield 97.16%, 97.09%, and 96.79%. As the ratio increases to 70:30, LSTM improves to 97.30%, slightly higher than BiLSTM (97.11%), while RF and NB remain competitive at 97.19% and 97.38%. Under the 80:20 split, BiLSTM achieves its peak performance of 97.43%, marginally exceeding LSTM (97.35%), whereas RF, NB, and XGB reach 97.23%, 97.23%, and 96.89%.

Although the 60:40 split achieves the highest Acc and F1-score, the 80:20 split is selected as a representative configuration due to its larger training set and its common use in practical ML settings. This choice reflects a balance between performance and training data availability, and this distinction is clarified to avoid potential confusion in interpreting the results.

Table 4 presents the Pre comparison across all models and data splits. The Fine-tuned DistilBERT consistently achieves the highest Pre, ranging from 98.86% to 99.07%, outperforming both DL and traditional ML counterparts. Among deep models, BiLSTM attains the best Pre of 98.39% under the 80:20 split, whereas LSTM shows greater fluctuation, dropping to 94.66% at 60:40 but improving to 98.34% at 70:30. In contrast, RF and XGB maintain stable Pre around 98% across all ratios, while NB trails slightly at approximately 97.2%.

Table 5 illustrates the Rec performance of the evaluated models. The Fine-tuned DistilBERT again demonstrates superior stability, maintaining nearly 99% Rec across all training–testing configurations. LSTM achieves its highest Rec of 98.58% under the 60:40 split, highlighting its sensitivity in phishing email detection, though it declines slightly to 97.36% at 70:30. BiLSTM and NB remain competitive with Rec values above 98% in certain ratios, whereas RF and XGB consistently perform within the 97–97.5% range.

Table 6 presents the F1-score comparison across all training–testing ratios. The fine-tuned DistilBERT consistently achieves the highest results—99.02%, 98.97%, and 98.91% for the 60:40, 70:30, and 80:20 splits, respectively. Under the 60:40 configuration, BiLSTM attains 97.91%, outperforming LSTM (96.59%) and surpassing RF, NB, and XGB, which record 97.72%, 97.69%, and 97.42%. When the ratio increases to 70:30, LSTM improves to 97.83%, slightly exceeding BiLSTM (97.71%), while RF and NB remain competitive at 97.75% and 97.92%. Under the 80:20 split, BiLSTM reaches its peak (97.94%) and marginally outperforms LSTM (97.89%), whereas the classical models range between 97.52% and 97.80%. Overall, DistilBERT demonstrates superior stability and balance between Pre and Rec across all partitions.

Table 3. Comparison of models by testing accuracy across training–testing ratios

Training–Testing Ratio	RF	NB	XGB	LSTM	BiLSTM	Fine-Tuned DistilBERT
60:40	0.97163	0.97092	0.96793	0.95638	0.97377	0.987742
70:30	0.97187	0.97377	0.96807	0.97301	0.97111	0.987077
80:20	0.97234	0.97234	0.96892	0.97348	0.97434	0.986317

Table 4. Comparison of models by testing precision across training–testing ratios

Training–Testing Ratio	RF	NB	XGB	LSTM	BiLSTM	Fine-Tuned DistilBERT
60:40	0.98184	0.97247	0.98018	0.94666	0.97559	0.990658
70:30	0.98103	0.97358	0.97594	0.98343	0.96923	0.989976
80:20	0.98164	0.97210	0.97583	0.97428	0.98391	0.988626

Table 5. Comparison of models by testing recall across training–testing ratios

Training–Testing Ratio	RF	NB	XGB	LSTM	BiLSTM	Fine-Tuned DistilBERT
60:40	0.97268	0.98133	0.96835	0.98588	0.98269	0.989756
70:30	0.97389	0.98482	0.97298	0.97328	0.98512	0.989375
80:20	0.97404	0.98406	0.97449	0.98360	0.97495	0.989526

Table 6. Comparison of models by testing F1-score across training–testing ratios

Training–Testing Ratio	RF	NB	XGB	LSTM	Bi LSTM	Fine-Tuned DistilBERT
60:40	0.97724	0.97688	0.97423	0.96587	0.97913	0.990207
70:30	0.97745	0.97917	0.97446	0.97833	0.97711	0.989675
80:20	0.97782	0.97804	0.97516	0.97892	0.97941	0.989076

Table 7. Comparison of models by testing loss across training–testing ratios

Training–Testing Ratio	RF	NB	XGB	LSTM	Bi LSTM	Fine-Tuned DistilBERT
60:40	0.17176	0.10481	0.05858	0.18076	0.00043	0.082204
70:30	0.16771	0.09959	0.06116	0.12888	0.00049	0.094846
80:20	0.17575	0.10087	0.06237	0.16532	0.00013	0.092091

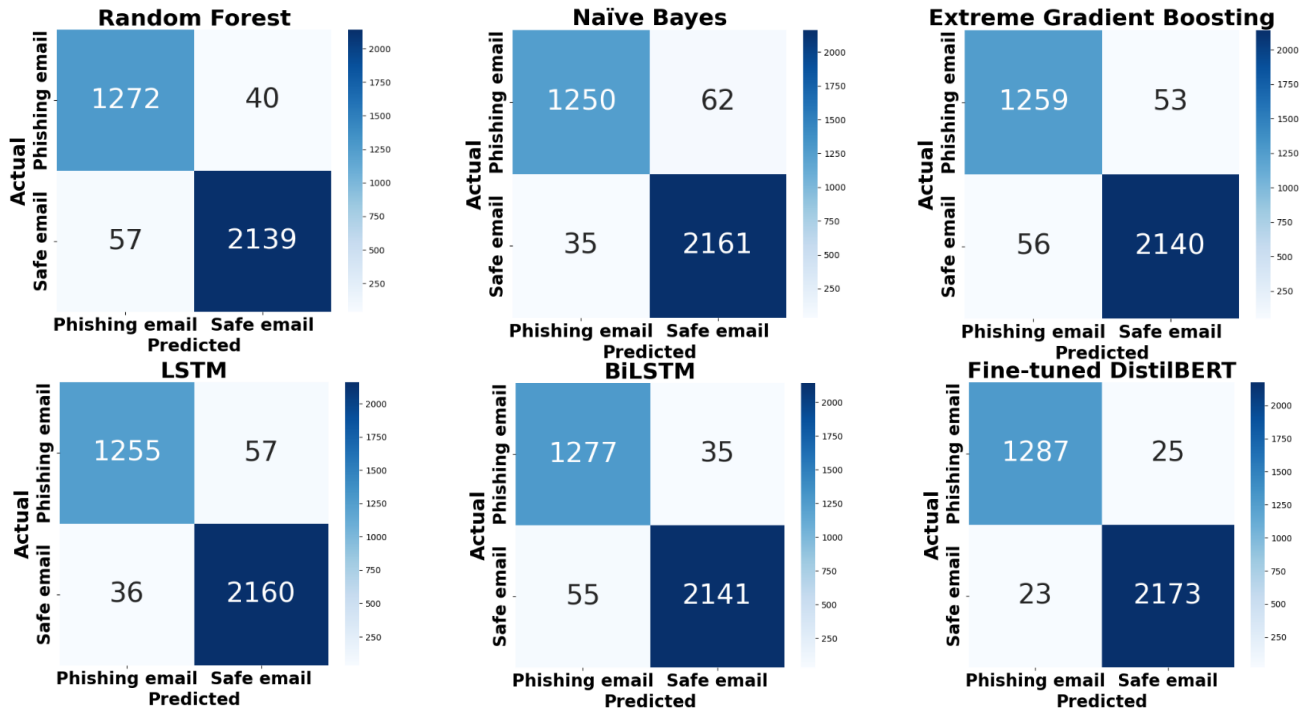

Figure 2. Confusion matrices (CM) of the evaluated models for phishing email classification, illustrating TP, TN, FP, and FN distributions

Table 7 presents the testing loss across models and data splits. As loss values are not directly comparable across different model families, they are interpreted as within-model indicators of optimization. BiLSTM achieves very low loss (0.00013–0.00049), while LSTM shows higher and more variable values (0.12888–0.18076). The fine-tuned DistilBERT maintains relatively low and stable loss (0.08220–0.09485). Among traditional models, XGB achieves lower loss (≈0.058–0.062) compared to RF (≈0.167–0.176) and NB (≈0.099–0.105). Accordingly, performance comparison is primarily based on Acc, F1-score, and AUC.

Figure 2 illustrates the CM, revealing clear performance differences between traditional ML and DL models. Among classical approaches, RF correctly classifies 1,272 phishing emails (TP) and 2,139 safe emails (TN) but still produces 40 false positives (FP) and 57 false negatives (FN). Naïve Bayes records 1,250 TP and 2,161 TN, with 62 FP and 35 FN, while XGB achieves 1,259 TP and 2,140 TN, misclassifying 53 safe emails as phishing and 56 phishing emails as safe. The DL models exhibit improved Acc: LSTM attains 1,255 TP and 2,160 TN, with 57 FP and 36 FN, whereas BiLSTM enhances Rec with 1,277 TP but incurs 55 FP and 35 FN. In contrast, DistilBERT delivers the strongest performance, correctly identifying 1,287 phishing and 2,173 legitimate emails, with only 25 FP and 23 FN. These results highlight the superior balance between sensitivity and specificity achieved by the transformer-based model compared with both traditional and recurrent architectures. Among the evaluated models, the fine-tuned DistilBERT shows lower FN counts, indicating stronger capability in detecting phishing emails.

In a security context, false negatives are more critical than false positives, as they correspond to phishing emails that bypass detection and reach end users. In contrast, false positives may reduce usability by incorrectly flagging legitimate emails, but do not directly expose users to attacks. Therefore, minimizing false negatives is particularly important in phishing detection. The relatively low number of false negatives observed for the DistilBERT model indicates its effectiveness in reducing potential security risks while maintaining acceptable usability.

Figures 3-5 illustrate the ROC curves of the fine-tuned DistilBERT for the 60:40, 70:30, and 80:20 train–test splits, all achieving AUC values above 0.998. Notably, the 80:20 configuration yields the highest AUC of 0.9991, confirming the model's superior discriminative capability across data partitions.

Figure 6 presents the training time under different train–test splits. The results show that the training time increases from 289.10 s (60:40) to 321.35 s (70:30) and 342.38 s (80:20), reflecting the expected growth in computational cost as the size of the training data increases.

The experimental findings highlight the superiority of DistilBERT over both traditional ML and DL baselines. Under the 80:20 configuration, it achieves the highest performance across all metrics—Acc 98.63%, Pre 98.86%, Rec 98.95%, and F1-score 98.91%—while maintaining a stable loss of 0.0921. From the confusion matrix, DistilBERT correctly identifies 1,287 phishing and 2,173 benign emails, misclassifying only 25 benign as phishing (FP) and missing 23 phishing (FN).

In comparison, BiLSTM, the best-performing recurrent model, reaches an Acc of 97.43% and an F1-score of 97.94%, but still falls short of DistilBERT's balance between sensitivity and specificity. Although it records the lowest loss (0.00013) at this split, DistilBERT's stable convergence and superior classification effectiveness make it the more reliable choice. Moreover, its ROC curves for the 60:40, 70:30, and 80:20 train–test splits all achieve AUC values above 0.998, with the 80:20 configuration yielding the highest AUC of 0.9991, confirming exceptional discriminative capability. Hence, DistilBERT is selected as the proposed model due to its consistently high detection performance, particularly under the 80:20 split, which best represents real-world generalization behavior.

In addition, preliminary experimental runs conducted prior to the final evaluation showed consistent performance trends across different runs, suggesting the stability of the reported results. Future work may further extend this evaluation using stratified k-

fold cross-validation or repeated random splits to enhance statistical robustness.

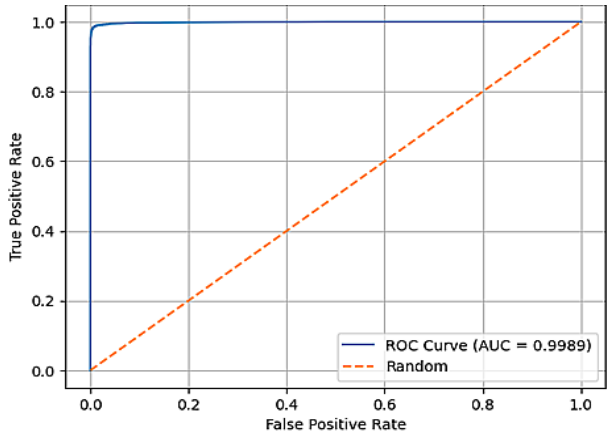

Figure 3. Fine-tuned DistilBERT ROC curve for the 60:40 testing split

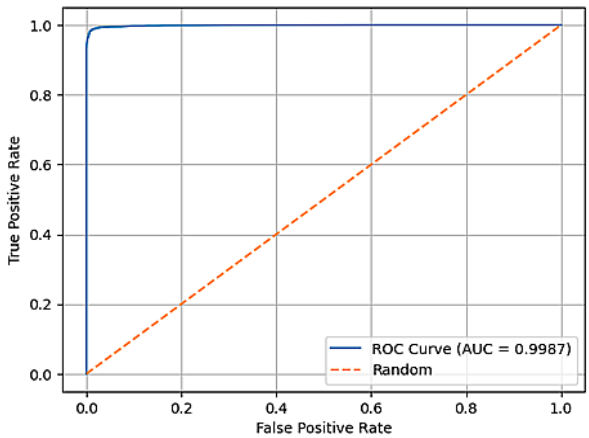

Figure 4. Fine-tuned DistilBERT ROC curve for the 70:30 testing split

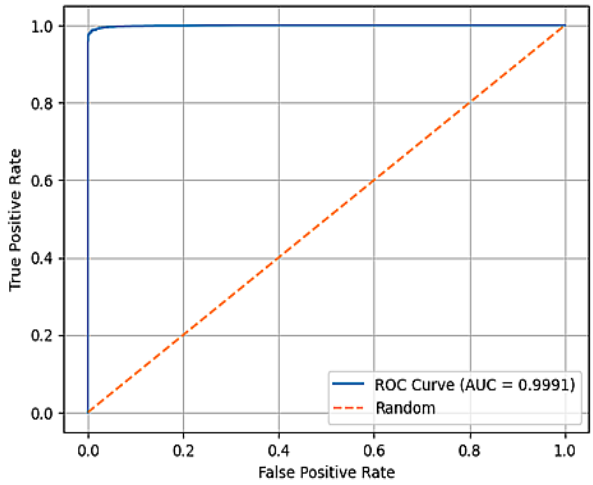

Figure 5. Fine-tuned DistilBERT ROC curve for the 80:20 testing split

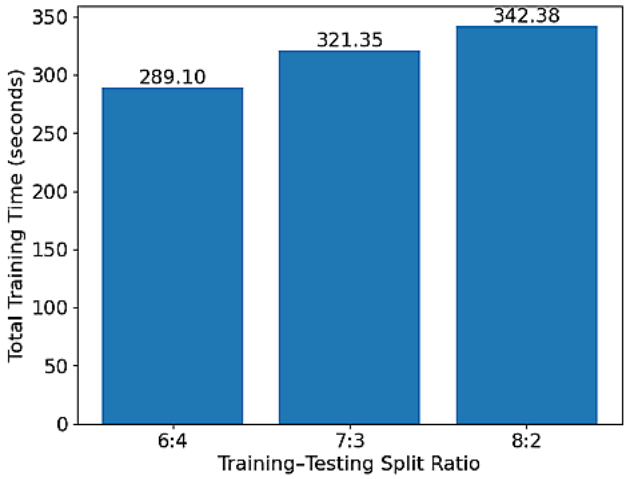

Figure 6. DistilBERT training time across different train–test splits

Table 8. Contextual summary of prior studies on phishing email detection

Study	Dataset and Size	Models	Reported Performance
[14]	4,591 emails (501 phish, 4,090 ham)	SVM, RF, XGB with TF-IDF, Word2Vec, Doc2Vec	Best Acc ≈ 98.8%
[15]	2,000 emails (Arabic, English, Chinese)	NB	Acc = 98.40%, FPR = 0.08%, FNR = 2.90%
[16]	15,000 emails (English, Spanish)	TF-IDF + LR or NB	Acc up to 98.5%
[17]	5,572 emails (4,572 ham, 1,000 phish)	BiLSTM, CNN, NB, *k*-NN, DT, SVM, AdaBoost, RF	BiLSTM Acc = 97%, F1-score = 88.89%; CNN Acc = 96.8%, F1-score = 87.45%; Traditional ML Acc < 93.6%
[18]	4,000 emails (2,000 ham, 2,000 phish)	LSTM, BiLSTM, SVM, GNB, DTC	LSTM-XGB Acc = 98.35%, F1-score = 98.24%; Others Acc < 97%, F1-score < 96%
[21]	33,727 emails (16,563 ham, 17,188 phish)	DNN+BiLSTM, LR, RF, RNN, CNN, LSTM	DNN–BiLSTM Acc = 98.69%, F1-score = 98.69%; Others Acc <96.39%
[22]	32,427 emails (9,001 ham, 9,138 harassment, 5,287 suspicious, 9,001 phish)	CNN, LR, SVM, NB, AdaBoost	CNN Acc = 98.43%, F1-score = 97.07%; Others Acc <89%, F1-score = 80.66%
[24]	7,286 emails (5,692 ham, 1,594 phish)	ALBERT, RoBERTa, BERT, DeBERTa, DistilBERT	BERT variants Acc = 98–98.8%, F1-score = 92–97%
Our Study	17,538 emails (10,980 safe, 6,558 phishing)	RF, NB, XGB, LSTM, BiLSTM, Fine-tuned DistilBERT	Fine-tuned DistilBERT: Acc = 98.77%, Pre ≈ 99.1%, Rec = 98.97%, F1-score = 99.02%, Loss = 0.0822, AUC = 99.91%, FP = 25, FN = 23, Time = 342.38 s (80:20 split)

4.2 Effectiveness of the proposed model in comparison with prior studies

Table 8 provides a contextual summary of prior studies on phishing email detection. As the experimental settings differ across studies (e.g., dataset size, language, preprocessing, and task scope), the reported results are not directly comparable.

In this study, the fine-tuned DistilBERT model is evaluated on a dataset of 17,538 emails under multiple train–test splits. Under the 80:20 configuration, the model achieves 98.63% Acc, 99.10% Pre, 98.97% Rec, and an F1-score of 99.02%, with 25 false positives and 23 false negatives. The model also achieves an AUC of 0.9991 with a testing loss of 0.0822.

These results demonstrate the effectiveness of the proposed evaluation framework and provide a consistent benchmark for comparing ML, DL, and transformer-based approaches under a unified experimental setting.

5. CONCLUSION

This study presents a systematic benchmark of machine learning (RF, NB, XGB), deep learning (LSTM, BiLSTM), and transformer-based approaches for phishing email detection. Experiments were conducted on a dataset of 17,538 emails (10,980 benign and 6,558 phishing) across three train–test ratios (60:40, 70:30, and 80:20) to comprehensively evaluate model generalization under varying data splits. The results consistently show that the fine-tuned DistilBERT outperforms all baselines, achieving its best performance under the 80:20 split with 98.77% Acc, 99.10% Pre, 98.97% Rec, and 99.02% F1-score, while maintaining only 25 false positives and 23 false negatives. The model also records a low loss (0.0822), an exceptional AUC (0.9991), and a total execution time of 342.38 s, demonstrating both robustness and efficiency. These findings confirm that lightweight transformer architectures such as DistilBERT can deliver state-of-the-art detection Acc with strong potential for scalable real-world deployment.

Despite its promising results, this study has certain limitations. The experiments relied on a single dataset of 17,538 emails, which may not fully capture the evolving diversity of phishing strategies or multilingual characteristics. Future research will therefore focus on expanding the dataset to larger, multilingual corpora, and on integrating semantic features from email headers and attachments to further enhance contextual understanding and detection robustness.

This study is conducted on a single-source dataset, which may limit generalization to other domains and real-world deployment scenarios. In practice, phishing attacks evolve over time and may vary across organizations and languages, leading to dataset shift and concept drift. As a result, the reported performance should be interpreted within the scope of the dataset used. Future work will consider cross-domain evaluation, multilingual datasets, and adaptive approaches to improve robustness under dynamic conditions.

ACKNOWLEDGMENTS

The authors sincerely thank the Editor-in-Chief, the reviewers, and the Associate Editor for their constructive and valuable feedback.

REFERENCES

[1] Anti-Phishing Working Group. (2025). Phishing Activity Trends Report, 4th quarter 2024. https://docs.apwg.org/reports/apwg_trends_report_q4_2024.pdf, accessed on Aug. 28, 2025.

[2] Verizon. (2024). 2024 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf, accessed on Aug. 28, 2025.

[3] Pal, S., Yadav, G., Jadidi, Z., Habib, A., Uddin, M.P., Karmakar, C., Shukla, S. (2026). Vulnerabilities in machine learning for cybersecurity: Current trends and future research directions. Journal of Information Security and Applications, 96: 104269. https://doi.org/10.1016/j.jisa.2025.104269

[4] Zhang, J., Wu, P., London, J., Tenney, D. (2025). Benchmarking and evaluating large language models in phishing detection for small and midsize enterprises: A comprehensive analysis. IEEE Access, 13: 28335-28352. https://doi.org/10.1109/ACCESS.2025.3540075

[5] Salahdine, F., El Mrabet, Z., Kaabouch, N. (2021). Phishing attacks detection a machine learning-based approach. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, pp. 0250-0255. https://doi.org/10.1109/UEMCON53757.2021.9666627

[6] Prasad, R. (2024). Phishing email detection using machine learning: A critical review. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, pp. 1176-1180. https://doi.org/10.1109/IC2PCT60090.2024.10486341

[7] Alhogail, A., Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. Computers & Security, 110: 102414. https://doi.org/10.1016/j.cose.2021.102414

[8] Magdy, S., Abouelseoud, Y., Mikhail, M. (2022). Efficient spam and phishing emails filtering based on deep learning. Computer Networks, 206: 108826. https://doi.org/10.1016/j.comnet.2022.108826

[9] Ling, F., Yang, H., Xiao, Y., Hu, L. (2024). Meta GPT-based agent for enhanced phishing email detection. In Proceedings of the 2024 14th International Conference on Communication and Network Security, Xiamen Shanghai, China, pp. 78-84. https://doi.org/10.1145/3711618.3711619

[10] Atawneh, S., Aljehani, H. (2023). Phishing email detection model using deep learning. Electronics, 12(20): 4261. https://doi.org/10.3390/electronics12204261

[11] Olea, C., Christensen, A., Fazio, L., Cutting, L., Lieb, M., Phelan, J., Wise, A., Tucker, H. (2025). Evaluating phishing email efficacy. In Proceedings of the 2025 Computers and People Research Conference, Waco Texas, USA, pp. 1-8. https://doi.org/10.1145/3716489.3728437

[12] Desolda, G., Greco, F., Viganò, L. (2025). APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users. Proceedings of the ACM on Human-Computer Interaction, 9(4): 1-33. https://doi.org/10.1145/3733049

[13] Saka, T., Vaniea, K., Kökciyan, N. (2025). SoK: Grouping spam and phishing email threats for smarter security. IEEE Access, 13: 54938-54959. https://doi.org/10.1109/ACCESS.2025.3555157

[14] Khalid, A., Hanif, M., Hameed, A., Ashraf, Z., Alnfiai, M.M., Alnefaie, S.M.M. (2024). Logitriblend: A novel hybrid stacking approach for enhanced phishing email detection using ml models and vectorization approach. IEEE Access, 12: 193807-193821. https://doi.org/10.1109/ACCESS.2024.3518923

[15] AbdulNabi, I.A., Yaseen, Q. (2021). Spam email detection using deep learning techniques. Procedia Computer Science, 184: 853-858. https://doi.org/10.1016/j.procs.2021.03.107

[16] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E. (2023). Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach. Applied Soft Computing, 139: 110226. https://doi.org/10.1016/j.asoc.2023.110226

[17] Zannat, R., Mumu, A.A., Khan, A.R., Mubashshira, T., Mahmud, S.R. (2023). A deep learning-based approach for detecting Bangla spam emails. In 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, pp. 1-6. https://doi.org/10.1109/ICECCME57830.2023.10252671

[18] He, D., Lv, X., Xu, X., Chan, S., Choo, K.K.R. (2024). Double-layer detection of internal threat in enterprise systems based on deep learning. IEEE Transactions on Information Forensics and Security, 19: 4741-4751. https://doi.org/10.1109/TIFS.2024.3372771

[19] Remmide, M.A., Boumahdi, F., Boustia, N. (2024). Toward a hybrid approach combining deep learning and case-based reasoning for phishing email detection. International Journal on Artificial Intelligence Tools, 33(05): 2450015. https://doi.org/10.1142/S0218213024500155

[20] McGinley, C., Monroy, S.A.S. (2021). Convolutional neural network optimization for phishing email classification. In 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, pp. 5609-5613. https://doi.org/10.1109/BigData52589.2021.9671531

[21] Krishnamoorthy, P., Sathiyanarayanan, M., Proença, H.P. (2024). A novel and secured email classification and emotion detection using hybrid deep neural network. International Journal of Cognitive Computing in Engineering, 5: 44-57. https://doi.org/10.1016/j.ijcce.2024.01.002

[22] Borra, S.R., Yukthika, M., Bhargavi, M., Samskruthi, M., Saisri, P.V., Akhila, Y., Alekhya, S. (2024). OECNet: Optimal feature selection-based email classification network using unsupervised learning with deep CNN model. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 7: 100415. https://doi.org/10.1016/j.prime.2023.100415

[23] Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

[24] Mehdi Gholampour, P., Verma, R.M. (2023). Adversarial robustness of phishing email detection models. In Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics, Charlotte NC, USA, pp. 67-76. https://doi.org/10.1145/3579987.3586567

[25] Patra, C., Giri, D., Kundu, B., Maitra, T., Wazid, M. (2025). Rhetorical structure theory-based machine intelligence-driven deceptive phishing attack detection scheme. Journal of Information Security and Applications, 94: 104184. https://doi.org/10.1016/j.jisa.2025.104184

[26] Opara, C., Modesti, P., Golightly, L. (2025). Evaluating spam filters and stylometric detection of AI-generated phishing emails. Expert Systems with Applications, 276: 127044. https://doi.org/10.1016/j.eswa.2025.127044

[27] Kirollos, A. (2024). Phishing email dataset. 2024. https://www.kaggle.com/kirollosashraf/datasets.