

Article

Hybrid AI-Based Detection of LLM-Generated Phishing Emails

Raghad Ghawa ^{1,*}  and Areej Alhogail ² ¹ College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia² Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; aalhogail@ksu.edu.sa

* Correspondence: rghawa@moi.gov.sa or raghadghawa1@gmail.com

Abstract

Phishing email attacks remain among the most common and damaging forms of cyber-crimes. With the emergence of generative artificial intelligence (Gen-AI), adversaries can automatically generate tailored, well-crafted phishing emails for each potential victim rather than relying on mass-distributed templates, thereby reducing the effectiveness of traditional detection systems. In this study, we propose a novel hybrid framework for detecting AI-generated phishing emails that leverages natural language processing (NLP), machine learning (ML), and deep learning (DL). The uniqueness of the proposed approach lies in the dual application of bidirectional encoder representations from transformers (BERT): (1) as an embedding model to extract deep contextual representations of email content; (2) as a fine-tuned classifier. Additionally, we integrate high-impact common-word features, derived from the best-performing classifier, to enhance contextual interpretation and improve discrimination between human-crafted and AI-generated emails. The framework was evaluated on a balanced dataset combining real and Gen-AI phishing emails and benchmarked across six ML/DL models—support vector machine (SVM), random forest (RF), logistic regression (LR), long short-term memory (LSTM) networks, BERT, and generative pre-trained transformer (GPT)—using standardized preprocessing, hybrid feature engineering, and optimized hyperparameters. Experimental results show that the BERT fine-tuned classifier, enhanced with the integrated common-word features, achieved the highest accuracy of 98%, outperforming all other models and demonstrating strong generalizability. This study demonstrates how integrating contextual cues and custom lexical signals can significantly improve the detection of AI-generated phishing content. Cybersecurity professionals, policymakers, and researchers can develop sophisticated and resilient defenses against emerging AI-enabled threats.

Keywords: phishing email detection; LLM-generated emails; hybrid AI-based detection; machine learning; deep learning; natural language processing; cybersecurity; secure computing



Academic Editor: Hung-Yu Chien

Received: 9 June 2026

Revised: 21 July 2026

Accepted: 24 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Phishing is one of the biggest threats in cybersecurity, and its effects on individuals, companies, and governments throughout the world are becoming worse. Phishing is a social engineering approach, which means it targets human vulnerabilities, not technological ones. It is incredibly powerful. It is hard to quit. Attackers typically claim to be trustworthy persons and trick their victims to disclose sensitive information such as login credentials, bank account details, or personal data [1]. Phishing attacks are a threat not just to individuals but also to enterprises with the danger of asset compromise, financial loss,

and reputational damage. The cost of cybercrime is estimated to be more than \$1 trillion per year globally and phishing is a significant part of the problem [2,3]. Security reports show that phishing attacks are widespread and increasingly sophisticated. The Verizon Data Breach Investigations Report (DBIR) reported that 36% of data breaches in 2023 involved phishing and social engineering was behind 83% of successful attacks. Email is still the number one attack vector, accounting for 96% of phishing campaigns [4]. Proofpoint's 2024 State of the Phish study says the financial impact of phishing attacks continues to grow. The Anti-Phishing Working Group (APWG) reported that the global average cost of a phishing assault climbed to \$85,000 in 2023 from \$65,000 in 2022. The APWG also reported a 20% year-on-year increase in phishing attacks, with 877,536 phishing attempts recorded in Q2 2024, of which 32.9% were targeting social media platforms. Business email compromise (BEC) frauds are also proliferating at an alarming rate with average phishing transfer requests of USD 89,520 in Q1 2024 [5,6].

A number of methods have been developed to identify phishing in response to these dangers. Traditional approaches include lexical and structural analysis of email content as well as URL and domain-based features [7]. As AI, ML, and DL are advancing, more advanced approaches for detecting phishing are introduced. Algorithms such as support vector machine (SVM), random forest (RF), and logistic regression (LR) have been demonstrated to be useful in categorizing phishing attempts. Deep neural networks (DNNs), long short-term memory (LSTM) networks, and transformer-based models such as bidirectional encoder representations from transformers (BERT) have enhanced the detection accuracy [8–10]. However, most prior studies have focused primarily on phishing emails generated by humans.

The rapid advancement of generative artificial intelligence (Gen-AI) introduces new challenges to cybersecurity. More advanced models such as GPT-3 and 4 are able to generate text that is highly coherent and contextually realistic, creating phishing emails close to legitimate communications that pass traditional content-based filters. Not only can LLMs generate text, but they have also shown broad capabilities across diverse domains such as financial decision making [11], further showing the versatility of LLMs, and the growing importance of understanding LLM behavior in specialized contexts, producing phishing emails that closely resemble legitimate communications and can evade traditional content-based filters. Although certain studies have examined the application of AI in cybersecurity, there is a notable deficiency in systematic investigations into the detection of phishing emails produced by Gen-AI, indicating a substantial research gap. To the best of our knowledge, this study is one of the few to systematically evaluate six ML/DL classifiers including traditional ML (RF, LR, SVM) and deep learning models (LSTM, BERT, GPT-2) on a large-scale, recent Gen-AI phishing dataset released in 2024. The study employs BERT in a dual role for contextual embedding extraction and fine-tuned classification, uses high-impact common-word features to achieve the best-performing detection model and adopts a distinct methodological approach that differs from prior research.

This study addresses this gap by proposing a novel approach to detecting phishing emails generated by Gen-AI. While prior studies have primarily focused on traditional phishing detection methods or human-crafted phishing content, limited attention has been given to AI-generated phishing attacks, which present unique linguistic and structural challenges. The novelty of this work lies in its comparative evaluation of multi-model ML/DL performance on both human- and AI-generated phishing datasets—an aspect largely unexplored in the existing literature. The main contributions of this study are as follows:

1. A systematic comparative evaluation of six ML/DL classifiers (SVM, RF, LR, LSTM, BERT, and GPT-based models) for detecting Gen-AI-generated phishing emails. All

models were evaluated under the same dataset, preprocessing pipeline and metrics, ensuring a fair and direct comparison across traditional ML/DL approaches.

2. A hybrid detection framework combining NLP preprocessing, TF-IDF lexical features, and BERT contextual embeddings, with targeted hyperparameter optimization. Within this framework, BERT is used in two ways: as a feature extractor for generating dense contextual embeddings and as a fine-tuned binary classifier—a dual design choice that proved central to achieving the best results. The framework also incorporates high-impact common-word features from the best-performing classifier to further improve discrimination between AI-generated phishing and legitimate emails.
3. Empirical validation on a large-scale, recent Gen-AI-specific phishing dataset (2024, 103,490 samples; Section 3.1.2), showing that the proposed framework reaches a best-model accuracy of 98% with consistent performance across all six classifiers and practical relevance to real-world enterprise email security systems (Section 5).

The remainder of this work is structured as follows. Section 2 describes related work. Section 3 describes the technique. This section discusses the dataset development, the feature engineering methods, and the ML/DL classifiers employed. The results and comparative analysis are shown in Section 4. The discussion of the findings is presented in Section 5, and Section 6 concludes the work and proposes directions for further research.

2. Literature Review

Phishing detection research has moved through three broad phases over the years: early rule-based and machine learning methods, more powerful deep learning approaches, and the more recent challenge of detecting phishing content generated by large language models. Each phase is reviewed below, along with its key limitations.

2.1. Traditional Machine Learning Approaches for Phishing Detection

The earliest detection systems used blacklists, heuristic rules, and classical ML algorithms to flag suspicious URLs, email headers, and lexical patterns. Rao and Pais [12] compared RF, DT, and SVM for phishing URL detection and found RF to be the strongest, reaching a 96.5% accuracy. Harikrishnan et al. [13] pushed accuracy higher by adding temporal URL features, and Sahingoz et al. [14] built a real-time system using RNNs on email text and URLs, achieving a 97.98% accuracy.

These approaches established useful baselines but share a common weakness: they depend on surface-level features and cannot capture the deep semantic patterns of Gen-AI-generated phishing emails, which are fluent and contextually convincing enough to slip past feature-based filters.

2.2. Deep Learning-Based Phishing Detection

Deep learning brought clear gains over classical methods. Ullah et al. [8] combined CNN, RNN, and attention mechanisms for zero-day phishing attack detection with a 98.74% accuracy. Thakur et al. [15] confirmed that CNN and RNN models outperform traditional ML consistently. Transformer-based models further pushed the performance: Atawneh and Aljehani [16] combined BERT with LSTM to achieve a 99.61% accuracy, and Alhogail and Alsabih [7] proposed a GCN-based classifier to achieve a 98% accuracy. Beyond phishing specifically, Rodriguez et al. [17] showed that BERT and RoBERTa can reliably detect AI-generated scientific text, and Prova [18] found BERT to be far more effective than classical ML at identifying AI-generated content. Taken together, these studies—especially those using BERT in dual or combined roles [16–18]—point to the value of rich contextual representations for text-based detection.

That said, most of these models were trained on manually crafted phishing datasets and have not been tested against Gen-AI-generated content, which limits how useful they are against today's more sophisticated threats.

2.3. Detection of LLM-Generated Phishing Content

Interest in detecting AI-generated content first grew in education and social media. Cingillioglu [19] built an SVM-based tool for identifying AI-written academic essays, Alamleh et al. [20] used RF and NN for the same purpose, Alghamdi and Alowibdi [21] developed classifiers for Arabic AI-generated tweets, and Manley [22] and Karimi [23] raised concerns about AI-assisted plagiarism. Outside text, multimodal fusion methods have also shown strong results in detecting AI-generated video content [24].

In cybersecurity, Das and Verma [25] used RNNs to generate targeted phishing emails and achieved around a 94% accuracy on the Enron dataset. Qachfar et al. [26] used GAN and BERT for synthetic phishing text detection and obtained an F1-score of 99.6%. Eze and Shamir [27] found that LLM-generated phishing emails carry distinct linguistic fingerprints that ML classifiers can learn to spot with a 98% accuracy. It is worth clarifying here that GAN-based work by Anand et al. [28] and Kamran et al. [29] focuses on URL-level adversarial generation and class imbalance correction—a technically different problem from the content-level email detection addressed in this study. On the threat side, Heiding et al. [30] and Chataut et al. [31] documented how GPT-4 and Claude are making phishing attacks harder to detect, while Gryka et al. [32] and Gholampour and Verma [33] trained classifiers to withstand adversarially generated samples. The broader dual-use risks of Gen-AI have been discussed widely by John [34], Yigit et al. [35], Krishnamurthy [10], Usman et al. [36], and Gupta et al. [37].

Despite all this progress, existing work is held back by small datasets, limited model variety, or a focus on URL structure rather than email content. To our knowledge, no prior study has carried out a full comparison of six ML/DL classifiers on a large-scale, recent Gen-AI phishing dataset (2024, 103,490 samples) using a dual-role BERT architecture with hybrid feature engineering, which is exactly the gap this study fills.

3. Materials and Methods

3.1. System Design and Implementation

3.1.1. Proposed Model

The purpose of this research is to construct a strong framework for detection of AI-generated phishing emails by utilizing ML, DL, and NLP approaches. This study explores the potential of ML and DL to efficiently identify phishing emails created by Gen-AI systems from real emails.

The complete architecture is shown in Figure 1. The received email is preprocessed first, and then, hybrid feature extraction is performed to detect influential linguistic and structural features.

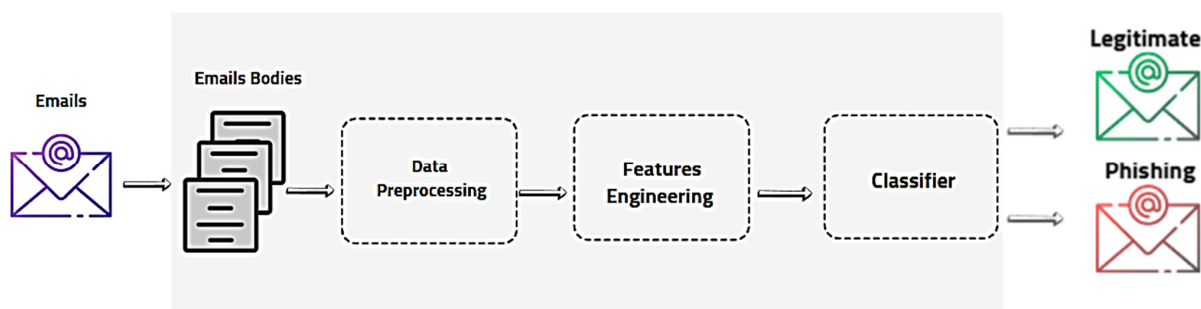


Figure 1. The architecture of the proposed phishing detection framework.

The extracted features are then passed through multiple ML/DL classifiers to predict whether the email is an AI-generated phishing attempt or a legitimate email.

To enhance clarity, Figure 2 presents the full methodological workflow, integrating NLP preprocessing, hybrid (manual + automatic) feature engineering, model training, and evaluation. This hybrid pipeline distinctly separates handcrafted lexical and structural features from automatically learned embeddings derived from transformer-based models. In this work, BERT was applied in two ways: as an extractor of rich contextual embeddings for each email content and as a classifier fine-tuned on the phishing datasets. Note that no automatic hyperparameter optimization approaches (e.g., grid search, Bayesian optimization) were utilized in this study; parameters were selected based on established literature defaults and refined through preliminary trials to ensure model stability and comparability across classifiers. The adoption of systematic automated optimization techniques is acknowledged as a direction for future work to further improve reproducibility and performance consistency.

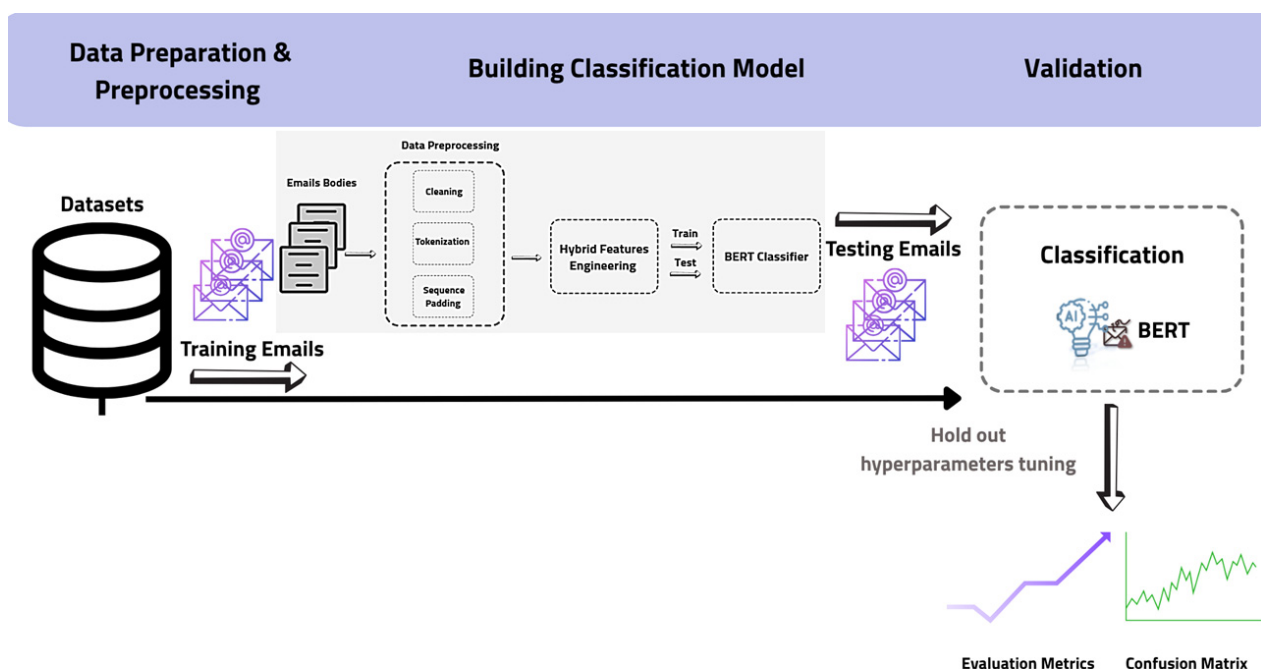


Figure 2. Hybrid feature engineering and classification pipeline for phishing detection.

3.1.2. Dataset Description

In this study, we use the Human-Generated and AI-Generated Email Dataset (2024) which consists of 103,490 email samples evenly distributed between the phishing and genuine classes. The dataset combines human-written emails from Nazario and the Nigerian Fraud collection [38], with AI-generated phishing emails generated by models such as ChatGPT and WormGPT that are quite good at generating convincing harmful emails. Each record includes fields such as sender, receiver, date, subject, body, URLs, a binary class label (1 = AI-generated phishing email, 0 = legitimate email), and source, allowing both linguistic and structural analysis. The dataset allows for rich feature extraction and supports both typical textual representations (e.g., TF-IDF) and deep contextual embeddings (e.g., BERT).

The dataset was separated into 80% training and 20% testing—a widely adopted split ratio in machine learning research [39]—using stratified sampling to guarantee that the two subsets have the same class distributions as the original dataset, which is important for robust model generalization. A separate validation set was not required in this study, as no automated hyperparameter optimization techniques (e.g., grid search, Bayesian opti-

mization) were employed. Instead, all hyperparameters were fixed to standard benchmark values that are commonly used in the relevant literature as detailed in Section 3.1.6. The test set was used solely for the final performance evaluation, without being used to make any tuning decisions to ensure an unbiased evaluation of model generalization.

To prevent data leakage, the same preprocessing, tokenization, and feature extraction processes were used to the training and testing sets. Specifically, all preprocessing operations—including TF-IDF vectorization and BERT tokenization—were fitted exclusively on the training set and subsequently applied to the test set, ensuring no information from the test partition influenced feature learning. Overfitting in DL models was further mitigated through dropout regularization (rate = 0.3) applied to all DL classifiers and conservative fixed-epoch training (5 epochs). The balanced dataset leads to a fair and reliable evaluation on all ML and DL methods presented in this study. For the supplementary experiments presented in Sections 4.3 and 4.4, the dataset was re-split into 70% training, 10% independent validation, and 20% held-out test subsets, with the validation subset used exclusively for early stopping and convergence monitoring.

3.1.3. Data Preprocessing

Data preprocessing was an important step to make the ML and DL models effective. The raw email content may contain noise like special characters, unnecessary spaces, and irregular formatting that may have a negative impact on feature extraction and model performance. Therefore, a structured preprocessing pipeline was developed to normalize the text input and prepare it for manual feature extraction such as TF-IDF and automatic embeddings such as BERT. Analysis of the preprocessed email corpus confirms that the vast majority of emails contain fewer than 128 tokens after stopword removal and stemming, making this threshold both efficient and representative without meaningful information loss.

Email Cleansing

The raw text was extracted from the body column and converted to string format using the `astype(str)` function. All characters were converted to lowercase using Python's `lower()` function. Non-alphanumeric characters (e.g., "!", "?", and "'") were filtered out using `str.isalnum()` inside a list comprehension. White spaces were also removed during filtering and tokenization. Figure 3 illustrates the function used and provides an example. It is acknowledged that the removal of special characters may discard stylistic cues relevant to phishing detection; future work will explore the incorporation of such features, including emoji frequency, punctuation density, and symbol-based urgency signals. Removing non-alphanumeric characters is standard NLP preprocessing. Notably, symbols such as excessive punctuation can carry weak but informative signals in traditional phishing emails. However, these patterns are less frequent by design in Gen-AI-generated phishing, where emails are designed to appear natural and legitimate, thus limiting the practical impact of symbol removal in this study.

Stopword Removal and Word Stemming

Stopwords such as "and", "the", and "is" were deleted using stopwords and the built-in English stopword list provided by NLTK (v3.8.1) (`words('english')`). Then, a word stemming was performed using the class `PorterStemmer()` from `nltk.stem`. The `stem()` function was applied to reduce each token to its root form. This reduced the size of the vocabulary and improved generalization by treating semantically similar words as the same. Figure 3 shows the function used with an example.

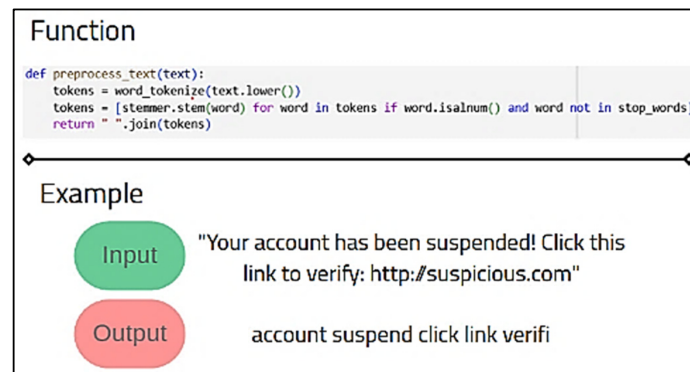


Figure 3. Emails clean and stopword removal and word stemming.

Tokenization

Tokenization was applied using the `word_tokenize()` function of the NLTK library, which splits the text into individual tokens (words). This was important to allow models to go over each word at a time. The DL models used the Keras `Tokenizer()` class to convert the cleaned text into integer sequences using `fit_on_texts()` and `texts_to_sequences()` methods. An example and the used function are shown in Figure 4.

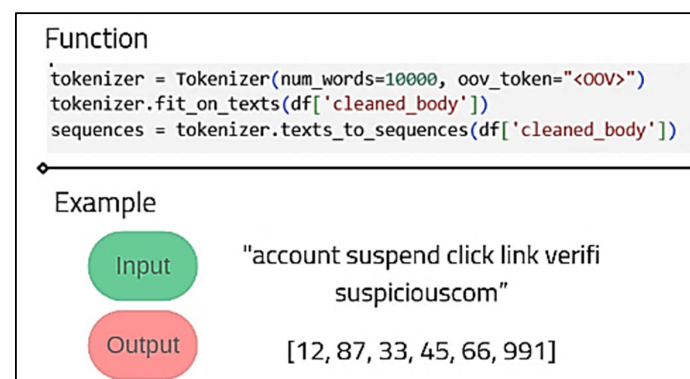


Figure 4. Emails tokenization.

Sequence Padding

To standardize the input length across all samples, sequences were padded to a maximum fixed length (`max_len = 200`) using the `pad_sequences()` method from Keras (`tensorflow.keras.preprocessing.sequence`). In order to provide the same input dimension to DL models, padding was performed by adding zeros at the end of the shorter sequences (`padding = "post"`). The utilized function and an example are displayed in Figure 5.

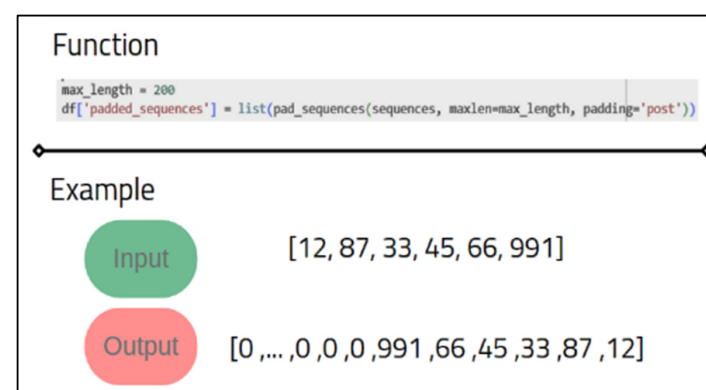


Figure 5. Emails sequence padding.

3.1.4. Feature Engineering

Feature engineering techniques were used to convert raw email data into structured numerical representations that could be used in a meaningful way by ML and DL models. The process resulted in a number of benefits including higher accuracy of classification, lower dimensionality, and more robustness against noisy data. The design of feature engineering was focused on capturing both lexical patterns and semantic linkages, allowing the models to differentiate between phishing and legitimate emails. Here, we used TF-IDF for the ML models and contextual embeddings for the DL models to improve performance and to ensure meaningful representation of the email content.

TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure that examines the importance of a word in a document with respect to a collection of documents (corpus). This strategy is useful in identifying domain-specific keywords [7,40], as it gives more weight to the words appearing frequently in a single document but infrequently in the dataset. In the feature engineering step of the ML models, the pre-processed email messages were converted into numeric vectors using TF-IDF, which can represent the importance of each word. The pre-processed email text was converted to a sparse matrix of TF-IDF scores using the Scikit-learn (v1.3.0) TfidfVectorizer (max_features = 5000). The discovered feature space contained 1218 distinct terms in 103,490 emails. As expected, terms like “update”, “account”, “secure”, and “http” received greater weights as these are words typically encountered in phishing emails.

BERT Embeddings

To capture richer contextual and semantic information, we utilized BERT embeddings. Unlike TF-IDF, which produces sparse keyword-based vectors, BERT generates dense (768-dimensional) representations that encapsulate the whole context of an email. This is especially useful for detecting AI-generated phishing emails, as BERT’s contextual representations can capture subtle linguistic patterns that distinguish them from legitimate emails [16,27].

Embeddings were extracted using the Bert-base-uncased model from Hugging Face Transformers. The [CLS] token output (outputs.last_hidden_state[:,0,:]) was used as a summary vector for each email. Input texts were tokenized and shortened to 128 tokens and processed in batches of 32 to ensure their efficiency on GPU hardware.

While the dimensions of BERT embeddings are not explicitly interpretable, they encode latent semantic and syntactic signals within a high-dimensional space. DL classifiers can leverage these contextualized representations to capture fine-grained distinctions between legitimate and phishing emails, thereby addressing the limitations inherent in traditional keyword-matching approaches. Furthermore, after evaluating classifier performance, high-impact common words identified by the best-performing model were integrated with the BERT embeddings to enhance contextual understanding and improve classification accuracy. This integration does not introduce representational conflict, as TF-IDF and BERT operate at complementary levels of text analysis: TF-IDF captures explicit lexical frequency patterns (e.g., high-frequency phishing keywords such as “verify”, “account”, and “urgent”), while BERT captures deep contextual and semantic relationships. The empirical improvement observed when incorporating these lexical features—from 0.97 to 0.98 accuracy—confirms that they supplement BERT’s contextual understanding rather than conflict with it. The feature fusion mechanism operates as follows. For the enhanced BERT classifier (Table 4), the [CLS] token embedding $e_{\text{BERT}} \in \mathbb{R}^{768}$, extracted from the final hidden layer of the fine-tuned BERT encoder, is concatenated with a compact keyword

feature vector f_{kw} representing the L2-normalized, TF-IDF-weighted presence of the highest-impact phishing-specific terms identified by the best-performing ML classifier (RF), as visualized in Figure 8. The resulting concatenated vector $h = [e_{BERT}; f_{kw}]$ is then passed to a linear classification head for binary classification. It is important to note that TF-IDF and BERT embeddings otherwise constitute two independent, parallel feature pipelines applied to ML and DL classifiers, respectively, and are not fused across classifier types.

3.1.5. Classification Algorithms for Phishing Email Detection

Classification algorithms are essential for identifying if an email is genuine or a phishing effort. In this study, we tested both traditional ML and sophisticated DL architectures, each providing different benefits in feature representation, generalization capacity, and effectiveness against AI-generated phishing content.

Among ML approaches, RF is one of the most popular methods [12,40], as it includes an ensemble learning mechanism that aggregates many decision trees to enhance robustness [7]. RF is successfully applied to analyze hand-crafted features from the email text, e.g., structural features and keyword frequencies. Although RF does well on classical phishing datasets, it tends to overlook significant grammatical distinctions in AI-generated phishing emails.

Another strong possibility is SVMs, which use an optimum hyperplane to separate classes maximally in high-dimensional feature fields. SVMs are good at handling huge sparse text data, but the need for considerable preparation and heavy computing can be a practical drawback.

LR predicts the chance of phishing with weighted text features as a simpler baseline. It is efficient enough for real-time use, but it cannot simulate deeper contextual and linguistic patterns that are often incorporated into generative phishing assaults.

LSTM networks are particularly powerful in the domain of DL as they can learn sequential and contextual relationships in the email content, which helps them to identify minor linguistic indicators that can differentiate phishing from authentic mails. However, their requirement for large amounts of training data and large processing resources can limit their scalability.

More sophisticated techniques are available in transformer designs such as BERT and GPT that use self-attention mechanisms to describe complicated contextual interactions. In particular, BERT was employed as a fine-tuned classifier and for embedding extraction. These models have performed better in detecting advanced phishing emails specifically created by Gen-AI, which are similar to human-written text. Transformers, when fine-tuned on domain-specific datasets, reliably obtain the best-performing model in terms of detection accuracy in this study.

We selected these models for our investigation, because they have been shown to be effective in phishing detection and are frequently used as benchmark classifiers in previous studies. These algorithms also represent complementary classes of strategies for text categorization. Support vector machine (SVM) and logistic regression (LR) capture discriminative patterns in high-dimensional sparse features, whereas Random Forest (RF) provides interpretability via feature-importance estimation. Long short-term memory (LSTM) networks can learn sequential dependencies that are relevant to contextual phishing content. Transformer models like BERT and GPT leverage self-attention to extract semantic and contextual cues, making them more effectively for analyzing Gen-AI emails that mimic human writing styles. The final BERT-based classifier was the most successful model including contextual embeddings and highly impactful common terms. This study combines traditional ML and modern DL techniques to provide a comprehensive assessment on their effectiveness and

also highlights the versatility of these algorithms in the context of the emerging problem of Gen-AI-driven phishing threats.

3.1.6. Model Training and Hyperparameter Optimization

After the definition of the model architectures, the ML and DL classifiers are constructed and trained to classify the emails as phishing or legitimate.

For the ML classifiers, TF-IDF feature vectors were used as input representations. We set up RF with 100 decision trees based on the Gini impurity criterion and trained LR with a maximum of 1000 iterations to ensure convergence. SVM was implemented with a linear kernel, which is well suited for high-dimensional sparse feature spaces. Although a full grid search was not performed, hyperparameters for all ML models were selected based on well-established defaults from the literature and verified through standard training convergence metrics, balancing accuracy with computational efficiency.

To reduce computational costs and focus training on the classification tasks, we used pre-extracted BERT embeddings instead of raw text for DL classifiers. The LSTM network has 128 memory units and a dropout layer (rate: 0.3) to mitigate overfitting, and it was trained for 5 epochs with the Adam optimizer (learning rate = 0.001) and a batch size of 32. The BERT-based classifier underwent fine-tuning on the phishing dataset, and its output was further enhanced by integrating high-impact words identified from the best-performing model. The BERT-based classifier was developed by freezing the pretrained BERT encoder and training only the additional dense layers. Specifically, it employs a 12-layer transformer encoder with 12 attention heads and a 768-dimensional hidden state. Two dense layers (256–64 ReLU units) with a dropout of 0.3 were trained before the sigmoid output to ensure robust generalization while preventing overfitting.

The GPT-based classifier, in contrast, follows a decoder-only transformer architecture that leverages causal self-attention. It was fine-tuned on phishing-text embeddings using a lightweight classifier head with dense and dropout layers preceding the final sigmoid output. Both transformer-based models were trained for five epochs with a batch size of 16. This architectural adaptation allows them to efficiently distinguish subtle stylistic and semantic differences between AI- and human-generated phishing emails.

By training only the new dense layers and freezing the pretrained embedding layers, this approach preserved the semantic knowledge captured during large-scale pretraining while enabling efficient fine-tuning for phishing detection tasks. Moreover, the models that worked best used both handcrafted lexical features and contextual embeddings, which improved their ability to capture subtle patterns characteristic of AI-generated phishing emails. The combination of contextual embeddings, a fine-tuned BERT classifier, and high-impact common words improved the model's ability to capture subtle patterns characteristic of AI-generated phishing emails. Table 1 presents the final configuration of key hyperparameters utilized across all classifiers.

Table 1. The key hyperparameters for ML/DL classifiers.

Classifier	Key Hyperparameters/Architecture Component	Optimized Value/Configuration
LR	Maximum Iterations (max_iter)	1000
	Solver	"lbfgs"
	Penalty	"l2"
RF	Number of Estimators (n_estimators)	100
	Criterion	"gini"
	Random State	"12"

Table 1. Cont.

Classifier	Key Hyperparameters/Architecture Component	Optimized Value/Configuration
SVM	Kernel	“linear”
	Probability Estimation	True
LSTM	Network Architecture	Layer 1: LSTM (128 units) Layer 2: Dense (64 units, ReLU) Layer 3: Dense (1 unit, Sigmoid)
	Optimizer	Adam
	Learning Rate (lr)	0.001
	Loss Function	“binary_crossentropy”
	Epochs	5
	Batch Size	32
	Regularization	Dropout (0.3)
	Callback	EarlyStopping
BERT	Classification Head Architecture	Layer 1: Dense (256 units, ReLU) Layer 2: Dropout (0.3) Layer 3: Dense (1 unit, Sigmoid)
	Optimizer	Adam
	Learning Rate (lr)	0.001
	Loss Function	“binary_crossentropy”
	Epochs	5
	Batch Size	16
	Regularization	Dropout (0.3)
	Callback	EarlyStopping
GPT	Classification Head Architecture	Layer 1: Dense (256 units, ReLU) Layer 2: Dropout (0.3) Layer 3: Dense (1 unit, Sigmoid)
	Optimizer	Adam
	Learning Rate (lr)	0.001
	Loss Function	“binary_crossentropy”
	Epochs	5
	Batch Size	16
	Regularization	Dropout (0.3)
	Callback	EarlyStopping

To ensure reproducibility, all ML experiments were conducted with `random_state = 42`. For DL models, global random seeds were fixed for the Python (v3.9.1) random module, NumPy (v1.24.3), and TensorFlow. All experiments were conducted with an NVIDIA A100-SXM4-80 GB GPU, Python 3.9.1, CUDA 11.8, TensorFlow 2.19.0, and HuggingFace Transformers (huggingface-hub 0.26.2).

3.1.7. Performance Evaluation

The proposed ML/DL model was evaluated for phishing email detection utilizing four key metrics—accuracy, precision, recall, and F-score (Wardhani et al. [41]). In this binary classification task, phishing emails are assigned the positive class label (label = 1),

while legitimate (human-written) emails are assigned the negative class label (label = 0). Accordingly, true positives (TP) refer to correctly identified phishing emails, true negatives (TN) refer to correctly identified legitimate emails, false positives (FP) refer to legitimate emails incorrectly classified as phishing, and false negatives (FN) refer to phishing emails incorrectly classified as legitimate.

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

Precision, also known as the positive predictive value, measures the proportion of emails correctly identified as phishing out of all those classified as phishing. Equation (2) presents the formula for precision:

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

Recall, also known as sensitivity or true positive rate, was the fraction of phishing emails properly identified by all phishing events. The recall formula is given in Equation (3):

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

The F1-score is very useful for the identification of AI-generated phishing emails in balanced datasets, as it gives a combined measure of recall and precision. The formula for the F1-score is shown in Equation (4) (Wardhani et al. [41]):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

These metrics are based on the confusion matrix, a table that shows the results of classification: true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP). Figure 6 demonstrates the form of a confusion matrix for binary classification (Wardhani et al. [41]).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 6. Confusion matrix.

To allow a robust evaluation and generalization, the dataset was split into 80% for training and 20% for testing using stratified sampling to preserve the proportional distribution of the phishing and authentic emails. This allowed us to verify that both sets are representative of the entire dataset. To ensure the consistency across the multiple feature representations, the splits were performed separately for the TF-IDF vectors and the BERT embeddings.

This evaluation methodology allows us to realistically and validly compare the ML and DL models and also to fully evaluate their performance in detecting AI-generated phishing emails. The next section introduces and discusses the experimental data obtained during our investigation. We thoroughly evaluate the effectiveness of the proposed methodology by using performance metrics and perform a comparative analysis to compare our findings with those reported in previous research.

4. Results

We evaluated the performance of different ML (RF, LR, and SVM) and DL (LSTMs, BERT, and GPT) models in detecting phishing emails generated by Gen-AI. The dataset is inherently balanced by design, thereby avoiding any bias toward a majority class. NLP techniques were then applied to improve the quality of textual representations by reducing textual noise and standardizing the structure of the input data. These preprocessing steps enhanced model efficiency and accuracy and contributed to the overall performance, which showed significant improvements in evaluation metrics (such as accuracy and F1-score).

To enhance the effectiveness of the models, we applied experiments with hybrid feature engineering techniques in two phases. We focused on transforming the preprocessed email content into understandable textual representations that could be processed by ML and DL models. We extracted lexical and semantic features from the email texts. The lexical features included surface textual features (such as word count, character count, and keyword frequency) often associated with phishing messages (e.g., “account update”, “emergency”, and “click here”). On the semantic side, BERT embeddings were first used to extract rich contextualized representations from email content, and subsequently, BERT was employed as a classifier fine-tuned on the training datasets. This dual use allowed the models to capture deeper linguistic and semantic patterns characteristic of AI-generated phishing emails. First phase: The ML models (RF, LR, and SVM) used sophisticated TF-IDF approaches for identifying the importance of words in a text context, while the DL models (LSTMs, BERT, and GPT) used word embeddings to obtain the semantic associations. In the second phase, we added to the fine-tuned BERT model common phishing-related terms as extra features to measure the effect of the added features on the model’s performance and further improve the identification of subtle manipulative cues, which are characteristic of generative phishing content.

4.1. Detection Results Without Feature Engineering

The baseline results without feature engineering are summarized in Table 2. Among the ML models, RF gave the best performance with an accuracy and recall of 0.95, followed by SVM and LR with accuracies of 0.93 and 0.92, respectively. In addition, DL classifiers performed better with BERT and GPT achieving 0.96 accuracies. These results demonstrate the intrinsic advantage of semantic feature representations in DL architectures compared to traditional ML models.

Table 2. Results of model performance without automatic feature engineering.

Classifier	Accuracy	Precision	Recall	F1-Score	Training Time (s)	Testing Time (s)
RF	0.95	0.85	0.95	0.90	52.36	1.92
LR	0.92	0.83	0.93	0.88	0.08	0.00
SVM	0.93	0.95	0.82	0.89	860.41	14.33
LSTM	0.95	0.85	0.94	0.90	543.22	10.04
BERT	0.96	0.92	0.97	0.94	98.73	1.01
GPT	0.96	0.91	0.96	0.93	99.12	1.02

4.2. Detection Results with Feature Engineering

When feature engineering was applied, performance improvements were observed across all models, as summarized in Table 3. RF, LR, and SVM benefited from TF-IDF-based lexical features; the accuracy of RF increased to 0.96 with a recall of 0.97, while LR achieved a 0.94 accuracy, and SVM, although its recall remained comparatively lower at 0.85, achieved the highest precision of 0.98. While DL models, including LSTMs, GPT, and BERT, leveraged rich semantic embedding. In particular, BERT was used in a two-step

process: first as a extractor of contextual embeddings from the email content and then as a fine-tuned classifier incorporating common phishing-related words as additional features. DL classifiers performed better than ML classifiers more substantially, with BERT achieving a 0.97 accuracy and an F1-score of 0.95, while GPT obtained similar results with a 0.97 accuracy and an F1-score of 0.94. Therefore, BERT was able to learn both high-level semantic regularities and precise lexical hints that are typical of AI-generated phishing emails.

Table 3. Results of model performance with automatic feature engineering.

Classifier	Accuracy	Precision	Recall	F1-Score	Training Time (s)	Testing Time (s)
RF	0.96	0.87	0.97	0.92	67.99	2.09
LR	0.94	0.86	0.95	0.90	0.12	0.00
SVM	0.95	0.98	0.85	0.91	1037.20	18.02
LSTM	0.96	0.87	0.96	0.92	687.43	11.80
BERT	0.97	0.94	0.98	0.95	107.91	1.04
GPT	0.97	0.93	0.97	0.94	106.47	1.03

The results show that hybrid feature engineering strategies (i.e., TF-IDF for ML classifiers and word embeddings for DL classifiers) are able to significantly improve the detection performance. Traditional ML models (RF and SVM) provided similar baselines in performance but were less efficient than DL models in detecting AI-generated phishing emails. RF reached a balanced trade-off with a precision of 0.87 and a recall of 0.97. The accuracy of the SVM model was enhanced to 0.95 and the precision was the best among the three ML models, increasing to 0.98. However, this higher accuracy comes at the expense of recall, with a relatively low recall of 0.85, which indicates that the model may be more conservative in identifying emails as phishing.

In contrast, DL classifiers, including LSTM, BERT, and GPT, outperformed ML classifiers. The addition of word embeddings for all DL models provided resulted in improved classification effectiveness. BERT outperformed all the classifiers. The model reached a 0.97 accuracy, a 0.94 precision, and a 0.98 recall with an F1-score of 0.95, showing its ability to recognize phishing communications, especially those created by AI. The results of BERT and GPT were similar with precision and recall scored of 0.97 and an F1 score of 0.94. This means that it is able to detect more phishing messages, but with a lower positive precision which is 0.94. The performance of LSTM model was comparable to RF where precision was increased to 0.96 and F1 score was increased to 0.92. These findings further validate the benefit of contextual embeddings to capture complex language and semantic patterns in phishing material. The accuracies of BERT and GPT are almost the same, but the learning behaviors are different. BERT's bidirectional attention better understands the context and has a higher recall. BERT performs well on syntactic cohesion and accuracy but sometimes misses implicit contextual clues. This is similar to the difference between encoder-based (BERT) and decoder-based (GPT) transformers.

From the confusion matrix of the BERT model, BERT accurately classified the majority of phishing emails, which shows the efficiency of the model in correctly detecting AI-generated phishing emails. The model misclassified about 207 communications as phishing or legitimate. Despite these flaws, the model exhibited good reliability in detection with an overall accuracy and recall of 0.98. Error analysis showed that false positives were often triggered by legitimate security notifications (e.g., password-reset emails) that share urgency-related linguistic cues with phishing messages. Typical false negatives were short, low-context emails that lacked strong phishing flags. These findings suggest that detection reliability can be further improved by incorporating metadata-based and pragmatic

We also recorded training and testing times before and after feature engineering. The BERT training time grew from 98.73 to 107.91 s, while the testing duration remained low with about one second. This modest overhead is a proof that the GPU acceleration is quite efficient against the quadratic temporal complexity of transformer models. The BERT model has a theoretical temporal complexity of $O(n^2)$ regarding the input sequence length (n) due to the self-attention mechanism used in the transformer design. This quadratic complexity can greatly affect training and inference times for extended sequences.

However, we used an input length limited to 128 tokens, and the use of GPU acceleration mitigated the impact of this complexity. Therefore, the addition of hybrid feature engineering, despite slightly increasing model complexity, did not lead to a significant degradation in execution time. This confirms that GPU utilization was effective in maintaining time efficiency, even for models with relatively high theoretical computational complexity. Despite the theoretical quadratic time complexity $O(n^2)$ of transformer self-attention, the system maintained practical scalability. By limiting input to 128 tokens and exploiting GPU acceleration, average inference time remained around one second per batch, demonstrating feasibility for large-scale enterprise deployment.

Overall, the evaluation confirms the superiority of DL classifiers, particularly BERT, in detecting AI-generated phishing emails. While ML models such as RF offered robust baselines, they were less effective in capturing the semantic and contextual patterns that characterize generative phishing content. By incorporating hybrid feature engineering and dual use of BERT, the fine-tuned BERT model achieved the highest performance (accuracy: 0.98; recall: 0.98; F1-score: 0.96), as detailed in Table 4. These results demonstrate the need of combining lexical and contextual semantic representations for effective phishing detection.

4.3. Performance Under Class Imbalance

To evaluate the robustness of the proposed framework under real-world conditions, we tested BERT + Common Words, the best-performing model under two imbalanced legitimate-to-phishing distributions: 90:10 and 95:05, reflecting the rare occurrence of phishing emails in actual enterprise email traffic. Results are presented in Table 5.

Table 5. Our model performance under imbalanced test distributions.

Classifier	90:10			95:05		
	Accuracy	Recall	F1-Score	Accuracy	Recall	F1-Score
BERT + Common Words	0.98	1.00	0.90	0.97	1.00	0.82

Compared to the balanced results reported in Table 4, F1-score decreases as imbalance increases, from 0.96 (balanced, Table 4) to 0.90 (90:10) and 0.82 (95:05). Notably, recall remains at 1.000 across both conditions, indicating that the model does not miss any phishing email even when phishing samples constitute only 5% of total traffic. The F1 drop is solely from a drop in precision that corresponds to an increase in false alarms on legitimate emails. This is an operationally acceptable trade-off in security critical deployments, where the cost of missing a phishing email is much higher than a false alarm.

4.4. Convergence Analysis

To justify the choice of five training epochs, Figure 9 presents the training and validation loss and accuracy curves for the proposed BERT + Common Words framework, obtained from the supplementary experimental setup (Section 3.1.2), which uses the same model architecture, hyperparameters, and training procedure as the main experiments. The convergence behavior is therefore representative of the full training pipeline. Both curves demonstrate consistent convergence: validation loss decreases steadily across all epochs

while validation accuracy stabilizes by epoch 5, with no signs of overfitting. These results verify that five training epochs are sufficient and provide an efficient training period for the proposed framework.

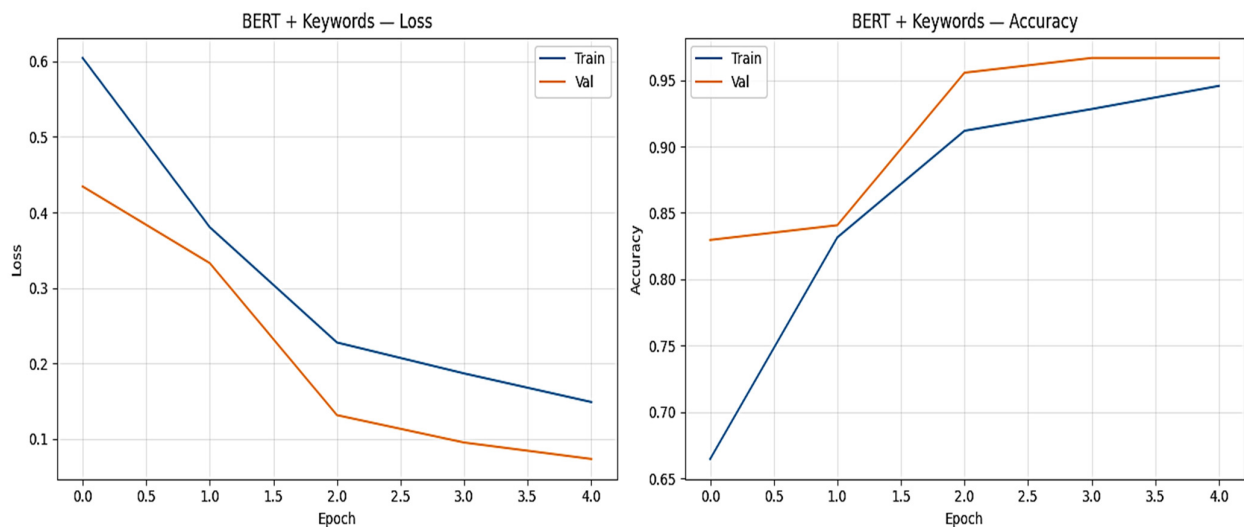


Figure 9. Training and validation loss and accuracy curves for the BERT + Common Words framework.

4.5. Ablation Study

To systematically validate the contribution of each component in the proposed hybrid framework and to justify the RF-based keyword selection strategy, we conducted a five-group hierarchical ablation study within the supplementary experimental setup described in Section 3.1.2, using the same model architecture, hyperparameters, and training procedure as the main experiments. The five configurations are defined as follows: G1 (TF-IDF Only) represents the traditional ML baseline; G2 (BERT Vanilla) uses BERT [CLS] embeddings without any lexical augmentation; G3 (BERT + Random Low-Frequency Words) serves as a control group to isolate the effect of keyword quality by replacing high-impact keywords with randomly selected low-frequency words; G4 (Common Words Only) uses the RF-identified lexical feature vector without BERT embeddings; and G5 (BERT + High-Impact Common Words) represents the complete proposed framework. Results are reported in Table 6.

Table 6. Hierarchical ablation study results.

Group	Classifier	Accuracy	Precision	Recall	F1-Score
G1	TF-IDF Only	0.98	0.97	0.99	0.98
G2	BERT Vanilla	0.89	0.93	0.84	0.88
G3	BERT + Random Low-Freq Words	0.84	0.93	0.73	0.82
G4	Common Words Only (No BERT)	0.61	0.56	0.97	0.71
G5	BERT + High-Impact Common Words	0.96	0.97	0.95	0.96

The results reveal three key findings. First, G5 achieves the highest F1-score among all deep learning configurations ($F1 = 0.96$), outperforming G2 by 10.8 percentage points, which quantitatively demonstrates the synergistic gain of fusing BERT contextual embeddings with RF-identified high-impact lexical features. Secondly, G3 ($F1 = 0.82$) is worse than G2 ($F1 = 0.88$), which indicates that adding random low-frequency words actively disrupts BERT's contextual representations, supporting that the improvement is from keyword quality and not augmentation. This validates the RF-based keyword filtering strategy. Third, G4

has the lowest overall F1-score ($F1 = 0.71$), indicating that lexical features alone are insufficient without BERT's contextual understanding. The performance trend $G4 < G3 < G2 < G5$ validates the complementary role of contextual embeddings and domain-specific lexical knowledge. It is interesting to note that G1 achieves higher performance on this evaluation subset owing to the well-known superiority of traditional ML models with limited training data, where BERT-based models need larger corpora to fully leverage contextual representations. As demonstrated in the main experiments (Tables 3 and 4), BERT + Common Words consistently outperforms the RF baseline on the entire dataset, thus supporting the validity of the proposed framework under realistic training conditions.

5. Discussion

The study reveals the efficiency of ML, DL, and NLP approaches in detecting Gen-AI-generated phishing emails. We constructed a full pipeline including preprocessing, hybrid feature engineering, and hyperparameter tuning, which allowed for identifying linguistic and structural characteristics common in phishing emails under robust and realistic evaluation settings. Among the tested models, the BERT-based classifier had the best performance with an accuracy and recall of 98% and an F1-score of 96%, indicating its better capacity to identify AI-generated phishing emails.

Recent work on AI-based phishing detection has made great strides. However, several investigations were limited by imbalanced datasets, which limited generalizability and raised bias. Some works (Anand et al. [28]; Kamran et al. [29]; Qachfar et al. [26]) tried to overcome this issue by using synthetic phishing emails generated by some models like GPT or CGANs. These initiatives helped to broaden training datasets but could also be artificially biased, affecting model learning. The data in our study were balanced, so we did not require resampling or synthetic data generation, giving a more realistic evaluation of the model's performance versus Gen-AI phishing material.

Unlike the research which mainly analyzed URLs or metadata (Anand et al. [28]; Kamran et al. [29]), our study emphasized the analysis of textual content with the help of NLP. Techniques, such as tokenization, text normalization, and sequence padding, dramatically enhanced semantic understanding, enabling deeper models such as BERT and GPT to learn phishing-specific linguistic cues that simple link analysis cannot easily learn. This gives a more complete and reliable detection method, especially for those generated with Gen-AI.

Hybrid feature engineering provided additional performance improvement. In the first phase, TF-IDF was used for ML classifiers, whereas for the DL models, contextual embeddings such as BERT and GPT were used. The second phase included some popular phishing-related phrases that were integrated into BERT and improved its performance in all measures. This method not only beats the classical ML classifiers (RF, LR, and SVM) but also surpasses the current hybrid techniques studied in the earlier studies. Table 7 indicates this comparative advantage. Our BERT-based model achieved an accuracy of 98% and a recall score of 98%, outperforming the closest comparable studies: an improvement of 7 percentage points over [26] (91%), 3 percentage points over [30] (95%), 4 percentage points over [42] (94%), and 1 percentage point over [32] (97%) in terms of accuracy.

Several recent studies (Gupta et al. [37]; John [34]; Yigit et al. [35]; Krishnamurthy [10]; Usman et al. [36]) have primarily emphasized conceptual frameworks without providing detailed quantitative metrics, limiting their comparability. Our work, by comparison, offers a well-documented empirical assessment over a range of criteria, like as accuracy, recall, precision, and F1-score, and hence a more rigorous and transparent standard.

Table 7. Comparison of related studies in terms of accuracy, F1-score, precision, and recall.

Reference	Accuracy	F1-Score	Precision	Recall
[26]	91%	94%	93%	95%
[27]	-	99%	-	-
[30]	95%	95%	95%	96%
[32]	97%	96%	99%	94%
[38]	94%	95%	-	-
Our model	98%	96%	95%	98%

Note. Results in this table are taken directly from the original publications and are included for contextual reference only. No direct cross-study performance comparisons are made, as each study used different datasets and experimental conditions. All model comparisons within this study are conducted on a unified dataset under controlled conditions, as reported in Tables 2–4.

Overall, this research shows that with a well-structured holistic approach that balances datasets, applies NLP preprocessing and utilizes hybrid feature engineering, the detection performance of phishing emails can be significantly improved. The resulting model can effectively detect the modern phishing emails produced by Gen-AI with a performance accuracy of 98% and with very few false negatives when detecting phishing emails. The suggested method is suited to be incorporated into smart email security systems either as a plug-in or together with other AI-based cybersecurity applications, providing a scalable and practical answer against the increasing threat of increasingly human-like phishing attempts. The comparison research shows that the combination of lexical (TF-IDF) and semantic (BERT embeddings) representations gives balanced interpretability and contextual awareness. For example, the TF-IDF captures explicit phishing cues, e.g., “verify”, “update”, “account”, whereas the BERT embeddings capture implicit manipulative tone and narrative coherence typical of Gen-AI emails. This synergy explains the better F1-scores of transformer-based models.

There are some limitations of this study that should be mentioned. First, all the experiments were conducted on a balanced benchmark dataset, which may not be realistic to the real-world email environment where phishing emails are much less than legitimate emails. Second, the evaluation was conducted on a single dataset, which limits the generalizability of the results. Third, hyperparameters were selected manually and a fixed-epoch training strategy was used without a separate validation set. Finally, input sequences were truncated at 128 tokens, which may result in lost context for longer emails. These boundaries directly motivate the future directions outlined in Section 6.

The improved performance of BERT is due to its self-attention mechanism that captures long-range contextual dependencies over the entire email text. Unlike ML models that use fixed lexical features, BERT dynamically weighs the importance of each word in the context of all other words, allowing it to detect subtle cues characteristic of LLM-generated phishing content, such as formal urgency language, domain-specific phrasing, and structured persuasive patterns, which keyword-based models simply cannot detect. The results of the ablation study in Table 6 further demonstrate the complementary nature of the two feature streams in the proposed framework. BERT’s contextual representations alone (G2, F1 = 0.88) yield moderate performance, while high-impact lexical features alone (G4, F1 = 0.71) are insufficient for accurate phishing detection. Importantly, G3 (BERT + Random Low-Frequency Words, F1 = 0.82) performs worse than G2, which indicates that the noise generated by keyword augmentation without quality control negatively affects BERT’s learned contextual patterns. On the other hand, G5 (full framework, F1 = 0.96) obtains the best F1-score among all deep learning configurations by providing BERT with a set of precisely targeted high-impact features identified using RF importance scoring, which helps the model to simultaneously capture contextual semantics and discriminative lexical signals. These results confirm the importance of the concatenation-based

feature fusion strategy and the RF-based keyword selection mechanism as integral and non-redundant parts of the proposed framework.

6. Conclusions and Future Work

In this study, we proposed a hybrid AI-based framework for detecting Gen-AI-generated phishing emails. The framework achieves an accuracy and recall of 98% and an F1-score of 96%, which compares favorably with related work in this area. The approach of NLP preprocessing, hybrid feature engineering, and hyperparameter tuning effectively differentiates AI-generated phishing emails from legitimate content even when the emails are highly fluent and human-like.

The framework was evaluated on a large, recent balanced dataset comprising 103,490 samples, and it showed consistent results for all six ML and DL classifiers with BERT performing the best. The hybrid feature engineering pipeline, combining TF-IDF, BERT contextual embeddings, and high-impact common-word features, improved detection accuracy while keeping computational costs reasonable, highlighting the strength of transformer-based models for real-world email security applications.

This study has several limitations worth noting when interpreting the results: (1) all experiments were performed on a balanced benchmark dataset that does not reflect real-world email environments where legitimate emails greatly outnumber phishing emails; (2) the evaluation was performed on a single dataset so the results may not be generalizable; (3) hyperparameters were chosen manually and not through automated search; (4) a fixed-epoch training approach was used; a separate validation set was introduced for the supplementary experiments but was not applied to the core evaluation; and (5) input sequences were truncated at 128 tokens which may cause the model to lose important context in longer emails.

Future directions are grouped into two tiers: short-term core revisions and long-term extended research. The short-term core revisions including the following: (1) evaluating the framework under real-world imbalanced conditions, including 90:10 and 95:5 legitimate-to-phishing ratios; (2) running McNemar's test and report mean $\pm$ standard deviation across multiple runs for stochastic DL models; (3) running ablation experiments comparing performance with and without symbol retention; (4) testing input lengths of 256 and 512 tokens; (5) validating the framework across multiple datasets and evaluating detection performance separately for phishing emails generated by different LLMs; and (6) carrying out a systematic quantitative and linguistic analysis of misclassified samples to better understand where the model fails.

The long-term extended research including the following: (1) integrating explainable AI (XAI) tools to help security analysts understand and trust model decisions; (2) optimizing the framework for real-time deployment with faster inference and better scalability; (3) testing across multilingual email datasets; (4) exploring automated hyperparameter tuning methods such as Bayesian optimization and grid search; (5) developing adaptive learning strategies that allow the model to keep up with new and evolving Gen-AI phishing tactics; and (6) exploring the effect of tuning the number of high-impact lexical features (e.g., TOP-K = 10, 25, 50, 100) on classification performance and investigating other approaches for feature fusion, such as weighted fusion and fusion mechanisms based on attention, to further enhance the synergy between BERT contextual embeddings and domain-specific lexical knowledge.

Finally, the proposed hybrid feature engineering approach yields meaningful performance improvement on BERT-based phishing detection. The above-mentioned limitations can be addressed in future studies to make the framework more practical, interpretable, and resilient against more sophisticated AI-based phishing attacks.

Author Contributions: A.A., conceptualization, supervision, project administration, writing, and funding. R.G., methodology, data curation, writing, and formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Ongoing Research Funding Program (ORF-2026-1700), King Saud University, Riyadh, Saudi Arabia, for financial support.

Data Availability Statement: The data supporting the findings of this study are publicly on Kaggle (Greco [38]): [<https://www.kaggle.com/datasets/francescogreco97/human-llm-generated-phishing-legitimate-emails/data?select=llm-generated>, accessed on 22 September 2025]. Additional information can be obtained from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, OpenAI ChatGPT (GPT-4) was used to assist in language refinement. The authors have independently reviewed, edited and taken full responsibility for all content presented in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alyozbaky, R.S.; Alanezi, M. Detection and analyzing phishing emails using NLP techniques. In *Proceedings of the 2023 IEEE International Conference on Human-Oriented Research in Agile Software Engineering (HORA)*; IEEE: Piscataway, NJ, USA, 2023. [CrossRef]
2. McAfee. The Hidden Costs of Cybercrime. 2024. Available online: <https://www.mcafee.com> (accessed on 19 November 2024).
3. Sadique, M.R.; Karim, A.; Dahal, K.; Alazab, M. Phishing email detection using machine learning classifiers and reinforcement learning. In *Proceedings of the 2020 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*; IEEE: Piscataway, NJ, USA, 2020; pp. 593–598.
4. Verizon. 2023 Data Breach Investigations Report. 2023. Available online: <https://www.verizon.com/business/resources/reports/dbir/> (accessed on 15 November 2024).
5. Gupta, B.B.; Arachchilage, N.A.G.; Psannis, K.E. Defending against phishing attacks: Taxonomy of methods, current issues and future directions. *Telecommun. Syst.* **2018**, *67*, 247–267. [CrossRef]
6. Anti-Phishing Working Group (APWG). Phishing Activity Trends Report, 2nd Quarter 2024. 2024. Available online: <https://www.apwg.org> (accessed on 15 November 2024).
7. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
8. Ullah, A.; Shah, R.A.; Nawaz, S.A.; Ahmad, N.; Malik, M.H. Enhancing phishing detection leveraging deep learning techniques. *J. Comput. Biomed. Inform.* 2024. Available online: <https://jcibi.org/index.php/Main/article/view/340> (accessed on 23 July 2026).
9. Benavides-Astudillo, E.; Fuertes, W.; Sanchez-Gordon, S.; Nuñez-Agurto, D.; Rodríguez-Galán, G. A phishing-attack detection model using natural language processing and deep learning. *Appl. Sci.* **2023**, *13*, 5275. [CrossRef]
10. Krishnamurthy, O. Enhancing cyber security enhancement through generative AI. *Int. J. Univers. Sci. Eng.* **2023**, *9*, 35–50. Available online: <http://www.ijuse.in> (accessed on 15 November 2024). [CrossRef]
11. Sheng, H.; Xuanqi, W.; Chang, Z.; Jiacheng, W.; Pingxia, D.; Yuwei, W. AIGC video detection based on the fusion of spatial-frequency-optical flow multimodal features. *J. Syst. Eng. Electron.* **2026**, 1–15. [CrossRef]
12. Rao, R.S.; Pais, A.R. Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Comput. Appl.* **2019**, *31*, 3851–3873. [CrossRef]
13. Harikrishnan, N.B.; Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Time split based pre-processing with a data-driven approach for malicious URL detection. In *Cybersecurity and Secure Information Systems: Challenges and Solutions in Smart Environments*; Springer: Cham, Switzerland, 2019; Volume 43, pp. 43–65.
14. Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine learning based phishing detection from URLs. *Expert. Syst. Appl.* **2019**, *117*, 345–357. [CrossRef]
15. Thakur, K.; Ali, M.L.; Obaidat, M.A.; Kamruzzaman, A. A systematic review on deep-learning-based phishing email detection. *Electronics* **2023**, *12*, 4545. [CrossRef]
16. Atawneh, S.; Aljehani, H. Phishing email detection model using deep learning. *Electronics* **2023**, *12*, 4261. [CrossRef]
17. Rodriguez, J.D.; Hay, T.; Gros, D.; Shamsi, Z.; Srinivasan, R. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2022; pp. 1213–1233. [CrossRef]
18. Prova, N. Detecting AI-generated text based on NLP and machine learning approaches. *arXiv* **2024**, arXiv:2404.10032.
19. Cingillioglu, I. Detecting AI-generated essays: The ChatGPT challenge. *Int. J. Inf. Learn. Technol.* **2023**, *40*, 259–268. [CrossRef]

20. Alamleh, H.; AlQahtani, A.A.S.; ElSaid, A. Distinguishing human-written and ChatGPT-generated text using machine learning. In *Proceedings of the 2023 Systems and Information Engineering Design Symposium (SIEDS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 154–158. [CrossRef]
21. Alghamdi, N.S.; Alowibdi, J.S. Distinguishing Arabic GenAI-generated tweets and human tweets utilizing machine learning. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 16720–16726. [CrossRef]
22. Manley, S. The use of text-matching software's similarity scores. *Account. Res.* **2023**, *30*, 219–245. [CrossRef] [PubMed]
23. Mohammadkarimi, E. Teachers' reflections on academic dishonesty in EFL students' writings in the era of artificial intelligence. *J. Appl. Learn. Teach.* **2023**, *6*, 105–113. [CrossRef]
24. Ma, T.; Du, J.; Huang, W.; Wang, W.; Xie, L.; Zhong, X.; Zhou, J.T. Agent Trading Arena: A Study on Numerical Understanding in LLM-Based Agents. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025*; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2025; pp. 5496–5514.
25. Das, A.; Verma, R. Automated email generation for targeted attacks using natural language. *arXiv* **2019**, arXiv:1908.06893.
26. Qachfar, F.Z.; Verma, R.M.; Mukherjee, A. Leveraging synthetic data and PU learning for phishing email detection. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy (CODASPY '22)*; ACM: New York, NY, USA, 2022; pp. 1–12. [CrossRef]
27. Eze, C.S.; Shamir, L. Analysis and prevention of AI-based phishing email attacks. *Electronics* **2024**, *13*, 1839. [CrossRef]
28. Anand, A.; Gorde, K.; Antony Moniz, J.R.; Park, N.; Chakraborty, T.; Chu, B.-T. Phishing URL detection with oversampling based on text generative adversarial networks. In *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1168–1177. [CrossRef]
29. Kamran, S.A.; Sengupta, S.; Tavakkoli, A. Semi-supervised conditional GAN for simultaneous generation and detection of phishing URLs: A game theoretic perspective. *arXiv* **2021**, arXiv:2108.01852.
30. Heiding, F.; Schneier, B.; Vishwanath, A.; Bernstein, J.; Park, P.S. Devising and detecting phishing: Large language models vs. smaller human models. *arXiv* **2023**, arXiv:2308.12287v2.
31. Chataut, R.; Gyawali, P.K.; Usman, Y. Can AI keep you safe? A study of large language models for phishing detection. In *Proceedings of the 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 548–554. [CrossRef]
32. Gryka, P.; Gradoń, K.; Kozłowski, M.; Kutyla, M.; Janicki, A. Detection of AI-generated emails—A case study. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*; Association for Computing Machinery (ACM): New York, NY, USA, 2024; pp. 1–8. [CrossRef]
33. Gholampour, P.M.; Verma, R.M. Adversarial robustness of phishing email detection models. In *Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics (IWSPA '23)*; Association for Computing Machinery (ACM): New York, NY, USA, 2023; pp. 1–10. [CrossRef]
34. John, N. Navigating the dual nature of generative AI: Opportunities and challenges in the 2024 cybersecurity arena. *Int. J. Mach. Learn. Cybern.* **2024**, *2*, 14–25.
35. Yigit, Y.; Buchanan, W.J.; Tehrani, M.G.; Maglaras, L. Review of generative AI methods in cybersecurity. *arXiv* **2024**, arXiv:2403.08701v2.
36. Usman, Y.; Upadhyay, A.; Gyawali, P.; Chataut, R. Is generative AI the next tactical cyber weapon for threat actors? Unforeseen implications of AI generated cyber attacks. *arXiv* **2024**, arXiv:2408.12806.
37. Gupta, M.; Akiri, C.; Aryal, K.; Parker, E.; Praharaj, L. From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy. *arXiv* **2023**, arXiv:2307.00691v1.
38. Greco, F. Human and LLM-Generated Phishing and Legitimate Emails Dataset. Kaggle. 2024. Available online: <https://www.kaggle.com/datasets/francescogreco97/human-llm-generated-phishing-legitimate-emails/data?select=llm-generated> (accessed on 22 September 2025).
39. Berrar, D. Cross-validation. *Encycl. Bioinform. Comput. Biol.* **2018**, *1*, 542–545. [CrossRef]
40. Zaki, M.J.; Meira, W.J. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2020.
41. Wardhani, N.W.S.; Rochayani, M.Y.; Iriany, A.; Sulistyono, A.D.; Lestantyo, P. Cross-validation metrics for evaluating classification performance on imbalanced data. In *Proceedings of the 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*; IEEE: Piscataway, NJ, USA, 2019; pp. 14–18. [CrossRef]
42. Lim, E.; Tan, G.; Hock, T.K.; Lee, T. *Turing in a Box: Applying Artificial Intelligence as a Service to Targeted Phishing and Defending Against AI-Generated Attacks*; GovTech: Singapore, 2021. Available online: <https://i.blackhat.com/USA21/Wednesday-Handouts/US-21-Lim-Turing-in-a-Box-wp.pdf> (accessed on 28 December 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.