Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076992

ARTICLE

Enhancing Phishing Email Detection Using DeepSeek-Generated Synthetic Data and DistilBERT Classification Models

Amani Al-Ajlan, Lama Almelaifi, Remas Alharbi, Shahad Al-Hussain, Fay Alfarraj, Najwa Altwaijry and Isra Al-Turaiki*

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

*Corresponding Author: Isra Al-Turaiki. Email: ialturaiki@ksu.edu.sa

Received: 30 November 2025; Accepted: 27 February 2026; Published: 08 May 2026

ABSTRACT: Phishing emails are an increasing threat to both individuals and organizations, demanding more sophisticated methods of detection beyond traditional blacklisting and heuristic techniques. One of the main challenges in phishing detection is the limited availability of high-quality phishing datasets. To address this issue, we use generative AI to create synthetic emails that help reduce class imbalance, improve model generalization, and overcome data scarcity. We employ a large language model, DeepSeek-7B-Chat, to generate realistic and context-aware phishing and non-phishing emails. Through prompt engineering and fine-tuning, the model produces diverse and modern phishing-style emails that strengthen phishing detection systems. The quality of the generated synthetic emails is evaluated using BERTScore, Self-Bilingual Evaluation Understudy (Self-BLEU), and Perplexity. Our results show that a fine-tuning model achieves the highest BERTScore F1-scores, indicating strong semantic similarity, while lower Self-BLEU and appropriate perplexity values reflect better diversity and fluency. To evaluate the usefulness of synthetic data in phishing detection, we train DistilBERT models on the original dataset, the synthetic dataset, and a combined version. The model trained on the combined dataset (original and prompt-based) achieves an accuracy and F1-score of 95% and 94%, respectively, on the original test set and of 93% and 93% on the combined dataset (original and prompt-based); the model trained on the combined dataset (original and fine-tuned) achieves an accuracy and F1-score of 95% and 95%, respectively, on the original test set and of 92% and 92% on the combined dataset (original and fine-tuned). Overall, models trained on combined datasets consistently outperformed the other approaches. Thus, our findings indicate that integrating original and synthetic data is more effective than training on either dataset alone and that integrating high-quality synthetic data generated by a large language model significantly improves phishing email detection, addressing key limitations in real-world data availability.

KEYWORDS: Phishing email detection; generative AI; large language model (LLM); Deepseek; synthetic data generation; DistilBERT; cybersecurity

1 Introduction

Phishing attacks have long been a significant cybersecurity threat, targeting individuals and organizations by exploiting their trust and tricking them into revealing sensitive information. These attacks typically occur via email, where attackers pretend to be trusted organizations to deceive victims into sharing personal details such as passwords, credit card numbers, or access to private systems. Regardless of advancements in cybersecurity, phishing attacks continue to be a significant threat that increases over time. These attacks often use deceptive language to seem legitimate and gain the user's trust. Phishing emails may contain links to malicious websites or attachments designed to install malware. To illustrate the magnitude of this

threat, the Anti-Phishing Working Group (APWG) [1] observed 989,123 phishing attacks in the fourth quarter of 2024, showing a consistent upward trend from 877,536 in Q2 and 932,923 in Q3 of the same year. According to the IBM Cost of a Data Breach Report 2025, phishing attacks account for approximately 16% of reported data breach incidents and represent a significant financial burden, with an average cost of $4.8 million [2]. Phishers use various methods, including email, social networks, text messages, and phone calls. An illustrative case is the 2018 FIFA World Cup phishing scam, where attackers sent fake emails offering free tickets to trick individuals into revealing personal data [3].

While modern cybersecurity technologies have made considerable improvements in detecting and blocking phishing attempts, the constantly changing nature of phishing techniques presents ongoing challenges. In recent years, attackers have become increasingly skilled at crafting emails that mimic legitimate correspondence, often bypassing traditional detection systems designed to flag suspicious content [4]. A significant concern in the phishing landscape is the emergence of generative AI as a tool for cybercriminals. Generative AI models, such as Generative Pre-trained Transformer (GPT) [5], were originally designed to assist with natural language processing (NLP) and text generation, enabling AI to create human-like text. However, when misused by cybercriminals, these models can generate phishing emails that are more personalized and convincing than previously possible. Attackers can use AI to craft emails that are tailored to the specific interests, behaviors, and vulnerabilities of potential victims, making it difficult for both recipients and existing email security systems to distinguish between legitimate and malicious communications [6]. The increasing sophistication of phishing attacks highlights the limitations of traditional phishing detection systems, which often rely on predefined rules or static machine learning models trained on historical datasets. These systems excel at identifying known phishing techniques, but they are less effective against new, AI-generated phishing emails that employ unique language structures. The static nature of these systems makes them vulnerable to the dynamic and adaptable nature of modern phishing emails, especially those enhanced by generative AI [7]. Considering these challenges, researchers and cybersecurity professionals are increasingly exploring the potential of generative AI as not only a tool for attackers but also a defense mechanism. Using AI to simulate phishing attacks, it becomes possible to study the unique features and patterns present in these emails, providing valuable insights into how they are structured. These insights can be leveraged to train more advanced detection systems that can recognize both traditional and AI-generated phishing attempts.

In this paper, we explore the role of generative AI in enhancing phishing email detection systems, particularly how AI-generated phishing emails can be used to improve the training of machine learning-based detection models. We utilize a large language model (LLM), rather than alternative deep learning approaches such as generative adversarial networks (GANs), for several reasons. First, LLMs excel at understanding and generating coherent, context-aware natural language, which is essential for producing realistic phishing and legitimate email content. Second, prior research demonstrates that LLMs achieve superior performance on natural language processing tasks, whereas GAN-based models are more effective primarily in image generation and certain non-textual fraud detection scenarios [8]. The main contribution of this work is the use of large language models to enhance phishing detection through the generation of training data that closely reflect real-world phishing scenarios. As attackers increasingly employ AI-driven techniques to craft phishing emails, we address this emerging threat by constructing a more comprehensive dataset that integrates six publicly available phishing datasets with newly generated synthetic data produced using DeepSeek-7B-Chat. This approach significantly increases data diversity and better captures the evolving nature of phishing tactics. DeepSeek-7B-Chat was selected due to its flexibility in creating realistic, meaningful, and diverse emails that align with modern phishing tactics. We utilized DeepSeek to generate a dataset using prompt-based generation and another through model fine-tuning, both simulating

phishing and legitimate emails. Subsequently, we evaluated the quality of the synthetic dataset and trained DistilBERT models using various combinations of the synthetic and original datasets. Finally, we evaluated the performance of the trained models to assess their accuracy and reliability in phishing detection. This work addresses a binary classification problem, in which synthetic datasets were generated for both phishing and legitimate emails. Although some of the publicly available datasets used in this study contain only phishing emails, these samples were integrated with phishing data from other sources, while legitimate emails were aggregated from complementary datasets to construct a unified binary classification model. By examining the characteristics of phishing emails, both those created by humans and those generated by AI, we aimed to develop a more adaptive and intelligent email detection system that adapts to new phishing techniques. Our findings aim to contribute valuable insights into improving phishing detection systems and advancing cybersecurity practices.

This paper is divided into six sections. Section 2 provides an overview of the related work on detecting phishing emails using various approaches. Section 3 outlines the methodology adopted in this research, including the dataset description and synthetic data generation. Section 4 discusses the results, including experimental settings and results. Section 5 presents the discussion, and finally, Section 6 presents the conclusions and future work.

2 Related Work

The problem of phishing emails has grown significantly as technology has advanced, requiring new solutions beyond traditional methods to effectively detect these risks. Gangavarapu et al. [9] aimed to identify the optimal feature set of emails for machine learning algorithms to detect phishing and spam emails by exploring various feature selection methodologies and evaluating them with multiple classifiers. They created three datasets, combining spam and ham emails from the SpamAssassin project [10] with phishing emails provided by Nazario [11]. In total, their datasets comprised 3844 samples. Five types of features were extracted: body, subject line, sender address, URL, and script-based features. Features were selected using a low-variance filter, high-correlation filter, feature importance-based filtering, minimum redundancy maximum relevance (mRMR), and principal component analysis (PCA). They evaluated several classification models, including Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), and a voting ensemble. Among them, the RF model with feature-importance-based selection achieved the best results, reaching 99.4% accuracy on the ham–phishing dataset and 98.4% on the ham–spam dataset, outperforming all other models.

Valecha et al. [12] addressed the problem of how to detect phishing emails using clues within them, either gain or loss persuasion cues. They focused on the efficiency of a model using only gain cues, loss cues, or a combination of the two. They developed three machine learning models, one for each type. The results showed that the models were better at detecting phishing emails with these persuasion cues, by 5% to 20%, than without. They used various methods to validate the results, including k-fold cross-validation, preprocessing, and different machine learning models, such as NB, LR (Logistic Regression), RF, and SVM. Valecha et al. [12] evaluated their approach using more than 18,000 phishing emails from the Millersmile dataset [13] and over 19,000 legitimate emails from the Enron corpus [14]. Their results showed noticeable gains: phishing email detection improved by 20% and 17% with gain-based persuasion cues with loss-based cues, respectively. Overall, their method achieved 96.52% accuracy, 98.53% precision, 94.20% recall, and a 15.55% improvement in F-score. The research highlights the impact of persuasion in phishing email detection and how it can help improve performance.

Alammar and Badawi [15] developed a phishing email detection system using machine learning techniques, focusing primarily on the RF algorithm. They used the WEKA toolkit to analyze the Phishing

Dataset for ML Feature Evaluation [16] and EmailHeaderAnomalyDetection [17], which contains 88,489 email samples with 126 features. Their results showed that RF achieved the highest accuracy at 99.03%, outperforming Naïve Bayes, which reached 90.13%. The study also applied several data preprocessing and feature selection methods to improve classification performance, including filters that remove misclassified samples. This research demonstrates the effectiveness of machine learning algorithms, particularly RF, in classifying phishing emails accurately.

Doshi et al. [18] proposed a dual-layer architecture for detecting phishing and spam emails using machine learning techniques and deep learning algorithms on an imbalanced dataset, comprising the Nazario [11] and SpamAssassin datasets [10]. Their approach combines features from the email body and other content, and they applied different traditional machine learning and deep learning methods, such as Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN). To address the class imbalance, the detection process is split into two layers, one focused on phishing and the other on spam. Their method achieved high performance, around 99.51% accuracy, 99.68% recall, 99.5% precision, and an F1-score of 99.52%.

Heiding et al. [19] investigated how large language models (LLMs) like GPT-4 can be used to create and detect phishing emails. They showed that LLMs can generate phishing emails automatically, examined the effectiveness of LLMs in identifying phishing emails, and discussed how these technologies can enhance cybersecurity training. Their study emphasizes the need for better phishing detection methods and suggests that LLMs can enhance cybersecurity training by personalizing content for user vulnerabilities.

Al-Subaiey et al. [20] presented a method for phishing email detection using a unified dataset [21] from six sources, totaling 82,486 emails (42,891 spam and 39,595 non-spam): Enron Email Dataset [14], Ling-Spam Corpus [22], SpamAssassin Public Corpus [10], CEAS 2008 Spam Challenge Corpus [23], Nazario Spam Dataset [11], and Nigerian Fraud Dataset [24]. They applied text preprocessing and feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec, as well as Support Vector Classifier (SVC), Multinomial Naive Bayes (MNB), and RF models. Explainable AI was integrated through Local Interpretable Model-Agnostic Explanations (LIME), which explains model predictions by identifying keywords or features influencing classifications. LIME slightly alters the input email text and observes how these changes impact the model's predictions, highlighting words such as "urgent," "account," and "verify", which are indicative of phishing. The SVC model with TF-IDF achieved the best performance with 99.19% accuracy, 99% precision, and an F1-score of 99%.

Atawneh et al. [25] tackled the problem of detecting phishing emails using various deep learning techniques. They explored CNNs, RNNs, long short-term memory (LSTM) networks, and bidirectional encoder representations from transformers (BERT) on a phishing dataset, comprising the Enron [14], SpamAssassin [10], and UCI phishing datasets, with a significant focus on feature extraction using NLP. The dataset comprised phishing and non-phishing emails, and the deep learning models were trained and tested using this dataset. They compared the BERT and LSTM models, achieving the highest accuracy of 99.61% and significantly outperforming other state-of-the-art techniques. A strength of this research is the comprehensive comparison across several deep learning techniques, demonstrating the importance of model selection when dealing with text-based cybersecurity issues.

Thakur et al. [26] provided a comprehensive review of phishing email detection techniques using deep learning models. The review examined methods such as CNNs, LSTM networks, and hybrid models, which utilize datasets like those of Enron [14] and SpamAssassin [10]. It highlights the effectiveness of these models, with accuracy rates exceeding 99%, emphasizing their ability to detect sophisticated phishing attacks. The paper discusses the variety of datasets used in phishing email detection, including publicly available and

self-generated datasets simulating phishing scenarios. The findings highlight the strengths of deep learning models in analyzing complex patterns and adapting to evolving phishing tactics.

Altwaijry et al. [3] proposed the use of one-dimensional CNN-based models (1D-CNNPD) for phishing email detecting, incorporating LSTM, GRU, and their variations. Two publicly available datasets were utilized: Spam Assassin [10] and Phishing Corpus [11]. The 1D-CNNPD with Leaky ReLU and Bi-GRU achieved higher performance compared to other advanced deep learning and machine learning phishing detection methods, achieving a precision of 100%, accuracy of 99.68%, F1-score of 99.66%, and recall of 99.32%. They demonstrated that integrating recurrent layers such as Bi-GRU with CNN-based models significantly improves detection performance.

Alhuzali et al. [27] presented a comprehensive evaluation of fourteen machine and deep learning methods to detect phishing emails using ten datasets, nine publicly available datasets, and a merged dataset. The findings demonstrate that deep learning models outperform machine learning models in both accuracy and robustness. Transformer-based models such as BERT and RoBERTa significantly outperform traditional methods, achieving a higher detection accuracy of approximately 99% on the balanced merged dataset. Overall, the study highlights the effectiveness of advanced deep learning models in detecting evolving phishing threats.

Hosseinzadeh et al. [28] presented a hybrid deep learning and Mountain Gazelle Optimizer (MGO) to detect phishing emails. They proposed a hybrid architecture that integrates BERT for contextual text representations, multi-head attention to emphasize important patterns, CNN layers for local feature extraction, and GRU units for sequential modeling, enabling the model to capture both detailed and high-level characteristics of email content. To fine-tune the model, the Mountain Gazelle Optimizer (MGO) is employed to automatically adjust hyperparameters and improve learning. Following evaluation on a publicly available phishing email dataset [29], the optimized model achieved 96.8% accuracy, 97.2% precision, 95.4% recall, and a 96.3% F1-score, outperforming baseline methods. These results demonstrate that integrating a transformer-based language model with adaptive optimization can significantly enhance the reliability of phishing email detection systems. Table 1 summarizes the related work.

Table 1: Summary of related works in phishing email detection.

Reference	Methodologies	Datasets	Results
[9]	Feature extraction (5 types), feature selection (low variance, high correlation, mRMR, PCA), Naïve Bayes, SVM, RF, and Voting Ensembles were applied	Three datasets created (3844 samples) from the SpamAssassin and Nazario datasets	RF: 99.4% accuracy (phishing), 98.4% (spam)
[12]	NLTK, Bi-LSTM, Word2Vec, k-fold cross-validation, NB, LR, RF, and SVM were applied	38,084 emails, Millersmile datasets, Enron corpus datasets	Accuracy: 96.52%
[15]	RF was the primary method applied for phishing detection, with Naïve Bayes for comparison purposes	Phishing Dataset for ML Feature Evaluation and EmailHeaderAnomaly-Detection with 88,489 instances and 126 attributes	RF accuracy: 99.03%; NB accuracy: 90.13%

(Continued)

Table 1 (continued)

Reference	Methodologies	Datasets	Results
[18]	TF-IDF, Exhaustive Feature Selection, Information Gain, Chi-Square Test, F-Classif, dual-layer architecture, ANN, RNN, and CNN were applied	8218 emails, Nazario's publicly accessible phishing corpus dataset, and SpamAssassin project	F1-score of 99.52%
[19]	Phishing emails were generated using GPT-4, Claude, ChatLLaMA, and Bard; interactions of 112 participants with spoofed emails were analyzed	Cambridge Cybercrime Dataset	Detection accuracy; from 60% to 85%; internal variance of up to 25% in misclassifying emails; only 7% of participants noticed unusual sender
[20]	TF-IDF SVC, MNB, and RF LIME were applied	Enron Phishing Email Dataset, CEAS 2008 Spam Challenge Corpus, Ling-Spam Corpus, Nazario Spam Dataset, Nigerian Fraud Dataset, and SpamAssassin Public Corpus	Accuracy of SVM with TF-IDF: 99.1%
[25]	CNN, RNN, LSTM, and BERT were applied using NLP for phishing detection	Enron, SpamAssassin, and UCI phishing datasets, with dataset diversity impacting performance	CNN Performance F1-score: 98.87 RNN Performance Accuracy: 98.58%, LSTM Performance Accuracy: 98.89% BERT-LSTM Performance Accuracy: 99.61%
[26]	Systematic literature reviews (SLRs)	Self-generated datasets simulating phishing scenarios, and domain-specific datasets from platforms	More than 99% accuracy
[3]	1D-CNNPD with LSTM, Bi-LSTM, GRU, and Bi-GRU were applied	SpamAssassin and Phishing Corpus	1D-CNNPD with Leaky ReLU and Bi-GRU: precision of 100%, accuracy of 99.68%, F1-score of 99.66%, and recall of 99.32%

(Continued)

Table 1 (continued)

Reference	Methodologies	Datasets	Results
[27]	Naive Bayes, Logistic Regression, SGD Classifier, XGBoost, Decision Tree, Random Forest, Extra Tree, CNN, RNN, LSTM, BERT, DistilBERT, RoBERTa, and GCN were applied	Ling, Enron, SpamAssassin, TREC, CEAS, Nazario_5, Nigerian_5, and merged dataset	BERT and RoBERTa: the highest accuracy of 99% on the merged dataset
[28]	BERT, multi-head attention mechanisms, CNN, GRU, and mountain gazelle optimizer (MGO) were applied	phishing email dataset from Kaggle	96.8% accuracy, 97.2% precision, 95.4% recall, and F1-score of 96.3%

As summarized in Table 1, existing phishing email detection approaches can be broadly categorized into traditional machine learning, deep learning-based models, and emerging LLM- and explainability-driven techniques. Traditional machine learning methods, particularly Random Forest (RF), have been widely adopted due to their robustness and interpretability. Studies such as [9,15] report high accuracy values exceeding 99%. However, these approaches rely heavily on basic feature extraction techniques, including low-variance filtering and PCA, and the absence of advanced models, which limit their adaptability to evolving phishing strategies.

Deep learning models have been introduced to overcome the limitations of traditional machine learning approaches. For example, study [12] incorporated persuasion cues using Word2Vec and Bi-LSTM, achieving an accuracy of 96.52%, but remained dependent on manual labeling and explored a limited range of deep architectures. More complex architectures were examined in [18], where a dual-layer architecture incorporating a CNN, RNN, and ANN framework addressed class imbalance and achieved an F1-score of 99.52%. Similarly, the authors of [25] evaluated CNN, RNN, LSTM, and BERT models, reporting an accuracy of 99.61% for BERT and LSTM. Recent research [3] has shown that combining CNNs with recurrent layers, particularly Bi-GRU, significantly improves phishing email detection, achieving an accuracy of 99.68% on two public datasets. Additionally, Alhuzali et al. [27] demonstrated that advanced deep learning models, especially transformer-based approaches like BERT and RoBERTa, outperform traditional machine learning methods in accurately and robustly detecting phishing emails across multiple datasets. Additionally, Hosseinzadeh et al. [28] demonstrated that a hybrid deep learning architecture combining BERT with CNN, GRU, multi-head attention, and adaptive hyperparameter optimization significantly improves phishing email detection performance compared to conventional approaches.

Recent studies have explored the use of large language models (LLMs) and explainable AI to enhance phishing detection. In [19], LLMs such as GPT-4 and Claude were employed to generate phishing emails, with detection accuracy ranging from 60% to 85%. Although this work demonstrates the potential of LLMs for data generation, it is constrained by a small sample size and reliance on a single dataset. Explainable AI techniques were investigated in [20], where LIME was integrated with classifiers such as SVC and RF, achieving an accuracy of 99.19% through the identification of key phishing-related keywords.

Overall, despite the strong reported results, many studies have used small datasets for evaluation purposes, which limits the generalizability of their results to new and unseen phishing attacks. Although

many methods report high accuracy, real-world performance is not often reflected. Common limitations across existing studies include imbalanced data [18], limited data diversity and small dataset sizes [12], and the use of outdated phishing datasets [3,9]. These limitations directly motivate the objectives of this research. Accordingly, this study aims to leverage generative AI to produce synthetic phishing data to (i) address data imbalance and limited dataset size, (ii) enhance dataset diversity to improve model generalization, and (iii) evaluate phishing detection models using larger and more recent datasets that better reflect contemporary phishing techniques. By explicitly targeting these gaps, the proposed research seeks to improve both the reliability and real-world applicability of phishing detection systems.

3 Materials and Methods

To evaluate the effectiveness of synthetic data in phishing email detection, we followed a methodology that includes dataset selection and preprocessing, synthetic data generation, synthetic data quality evaluation, classification model training, and model evaluation, as shown in Fig. 1.

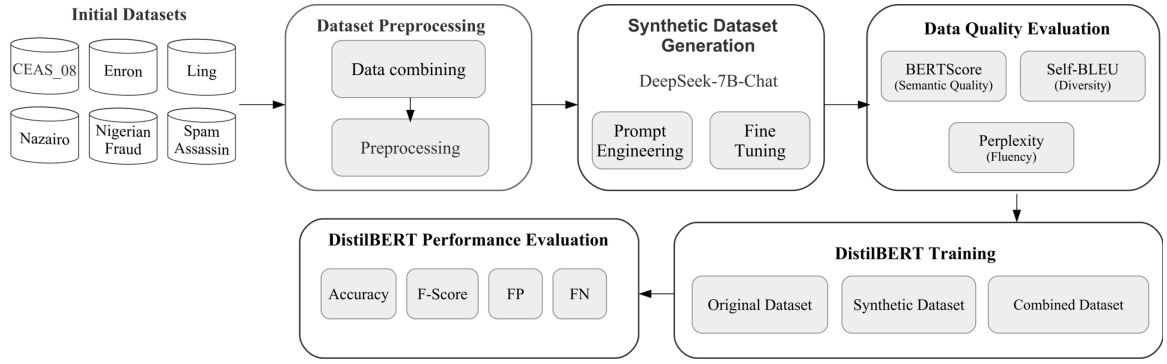

Figure 1: The workflow of the methodology.

3.1 Dataset Selection and Preprocessing

Dataset preprocessing is a critical step to ensure that the data used are clean, well-prepared, and ready for effective model training and evaluation. The process begins by retrieving publicly available phishing datasets from Kaggle [21]. This dataset combines six benchmark email phishing datasets into a single resource for analysis. As shown in Table 2, the initial datasets are CEAS_08 [23], Enron Corpus [14], Ling Spam [22], Nazairo [11], Nigerian Fraud [24], and SpamAssassin [10]. The final dataset includes a total of 82,486 emails, consisting of 39,595 legitimate (non-phishing) emails and 42,891 phishing emails. Table 3 shows examples of the emails in the dataset.

Table 2: Summary of dataset.

Dataset	# of Non-Phishing Emails	# of Phishing Emails	Total
CEAS_08	17,312	21,842	39,154
Enron	15,791	13,976	29,767
Ling	2401	458	2859
Nazairo	0	1565	1565
Nigerian_Fraud	0	3332	3332
SpamAssassin	4091	1718	5809
Total	39,595	42,891	82,486

Table 3: Example emails from the original dataset.

Label	Subject and Body of the Email
Phishing	"Subject: ID:19346 The world's largest online prescription-free apothecary Body: The ultimate convenience store in drugs, brought to you in just one click! Select from thousands of prescriptions. drugs to be delivered right to your doorstep. - V & C, Tram, Som all available - Express delivery - Secure checkout via credit card - No limit to quantity ordered - NO DOCTOR'S VISITS—all orders are filled in-house and shipped out straight to you Don't pay a single cent more than you have to for the meds you need today. Click here: www.outgoeffmedical.com"
Non-Phishing	"Subject: Wekalist Digest, Vol 60, Issue 10 Body: Send Wekalist mailing list submissions to iuxxdkqk@list.scms.waikato.ac.nz To subscribe or unsubscribe via the World Wide Web, visit https://list.scms.waikato.ac.nz/mailman/listinfo/wekalist Or, via email, send a message with subject or body 'help' to ohwwlejs-pbfotbp@list.scms.waikato.ac.nz. You can reach the person managing the list at pobypuhs-hvhzj@list.scms.waikato.ac.nz When replying, please edit your Subject line so it is more specific than "Re: Contents of Wekalist digest..." Today's Topics: 1. RE: Weka 3.5.7.exe installation problems (Danilovich, Yann) 2. Question (bug?) (Uday Kamath) 3. Event Sequence Mining (Budziak, G.) 4. RE: Question (bug?) (Cristina Torres) 5. RE: Where can i find algorithms explanation (Nanda, Subrat (GE, Research))"

Each source dataset has different features, as shown in Table 4. Due to these differences, the datasets required cleaning and reorganization to align with the requirements of our study. We focused on extracting the subject, body, and label features, which are essential for understanding the content of the emails and their classification as phishing or non-phishing. All other features (sender, receiver, date, and URLs) were removed to make the dataset more focused and efficient for our purposes.

Table 4: Comparison with the original email datasets.

Dataset/Features	Sender	Receiver	Date	Subject	Body	Label	URLs
CEAS_08	✓	✓	✓	✓	✓	✓	✓
Enron				✓	✓	✓	
Ling				✓	✓	✓	
Nazairo	✓	✓	✓	✓	✓	✓	✓

(Continued)

Table 4 (continued)

Dataset/Features	Sender	Receiver	Date	Subject	Body	Label	URLs
Nigerian_Fraud	✓	✓	✓	✓	✓	✓	✓
SpamAssassin	✓	✓	✓	✓	✓	✓	✓

We then cleaned the dataset to remove duplicates and incomplete entries and split it as follows for model development and evaluation: 70% for a training set, denoted as $D_{OrigTrain}$; 15% for a testing set, D_{Test}; and 15% for a validation set, D_{val}. This split ensured sufficient data for training, effective hyperparameter tuning, and an unbiased evaluation of the model's performance on unseen data.

3.2 Synthetic Data Generation

We aimed to create synthetic data to improve the ability of phishing detection models in identifying phishing threats. Two synthetic datasets were created using Deepseek 7b-chat LLM [30]: one via prompt-based generation and another through model fine-tuning. Both simulated phishing and legitimate emails, comprising 20,000 emails each. Both approaches were guided by predefined scenarios based on real-world contexts like account security, payments, and deliveries, as presented in Fig. 2. These scenarios reflect the most commonly observed phishing attacks reported in industry threat intelligence reports [1,31]. Each scenario was adapted to produce both phishing and non-phishing emails, helping the model capture subtle differences in intent and tone. This scenario-driven method reflects current cybersecurity practices, where controlled simulations are used to model realistic threats and improve detection systems [32]. To ensure privacy and variability, we applied Named Entity Recognition (NER) [33] and the Faker library [34] to replace any sensitive or identifiable information in the generated data.

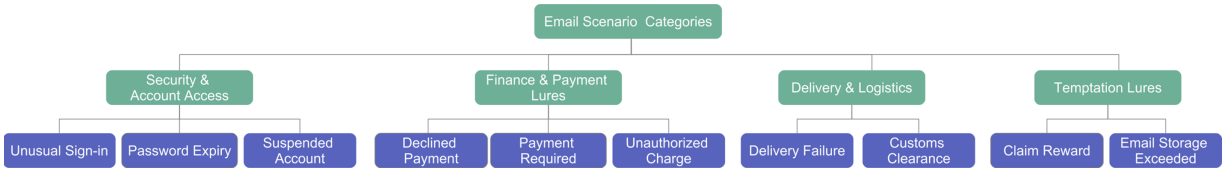

Figure 2: Synthetic dataset generation scenarios.

3.2.1 Prompt Engineering for the LLM

Prompt engineering involves designing structured input prompts to guide a pre-trained model to perform specific tasks without modifying its internal parameters [5]. In this study, we employed DeepSeek-7B-Chat [30], an open source instruction-following tasks model that supports chat-style interaction, as well as quantization (like 8-bit), which makes it easier to run on limited GPU resources. Additionally, there is currently limited research available on DeepSeek, which presents an opportunity to explore its capabilities in text generation. We utilized prompt engineering to guide Deepseek-llm-7b-chat to generate realistic emails without additional training. Around twenty thousand emails were generated, equally balanced between phishing and non-phishing emails across different scenarios. This approach involved the following steps:

- Few-Shot Prompting: We used two-shot prompting, which involves supplying the model with two input–output examples before issuing the main instruction. These demonstrations help the model understand structural patterns, desired tone, and domain-relevant subtleties. In our study, this approach

led to a notable improvement in both the plausibility and diversity of generated phishing and non-phishing emails. As shown in Table 5, the two-shot phishing email includes social engineering tactics such as urgency, implied consequences, and a spoofed support contact—hallmarks of realistic phishing content. The two-shot non-phishing email, in contrast, adopts a clear, informative tone and correctly directs users to a legitimate password update source. These improvements are consistent with previous research showing that few-shot prompting enhances the model's ability to mimic domain-specific language patterns and intentions more reliably than zero-shot methods [35,36].

- For each scenario, we crafted two example emails corresponding to the desired label (phishing or non-phishing) to guide the model in generating similar, realistic emails. In our study, we opted to craft the seed examples to ensure that all prompts consistently reflect common fields and conventions observed in public corpora, and that the few-shot examples follow a uniform format suitable for prompting. Using crafted examples, we also intentionally avoided personal identifiers and organization names, helping reduce potential privacy and legal risks. It is important to note that these few-shot examples are used only as minimal demonstrations to control formatting and contextual alignment, rather than as training data.
- Controlled Output: This approach encouraged the model to mimic the tone, intent, and structure of the provided examples, generating coherent and contextually appropriate emails.
- Postprocessing with NER: We applied NER to the generated text and used the Faker library to replace all named entities and sensitive content.
- Balanced Generation: A fixed number of emails per scenario were generated for both phishing and non-phishing classes.

Table 5: Example emails generated using two-shot prompt engineering.

Category	Example Prompt Structure	Example Generated Email
Phishing	"Write a phishing email pretending to be an email provider, warning that the user's password will expire soon. Here are two example emails: [Subject: Account Access Suspended Dear Customer, We detected unusual activity in your account and have temporarily suspended access for your protection. To restore your access, please verify your information immediately by clicking the link below: [Verification Link] Failure to act within 24 h may result in permanent account suspension. Thank you, Security Team], [Subject: Verify Your Email Password Dear User, Your email password is set to expire in 12 h. For uninterrupted access, please update your credentials using the secure link below: [Update Password] Regards, IT Support] — Now write a similar realistic email: Subject:"	"Subject: Important: Update Your Login Credentials Immediately! Body: Hello, great, This message is to inform you that we have discovered potential security breaches on our platform. As part of our ongoing commitment to keeping all customers' data safe and secure, we request immediate action from you regarding this matter. In order to protect your account, we need you to change your login credentials as soon as possible. Please follow the steps below: 1. Click on the following link: http://townsend.info/posts/categoryterms.jsp 2. Enter your current email address. 3. Choose a new password with at least one uppercase letter, lowercase letter, number, and symbol. 4. Re-enter your new chosen password to confirm. 5. Click 'Submit'. Should any issues arise during this process or if you feel unsure about updating your details, do not hesitate to reach out to our customer support team, who will assist you through these necessary changes—they can be reached via phone001-705-935-3048 or e-mail hunteralec@example.com. We apologize for any inconvenience caused, but assure you it's only done to keep your account fully protected. Best Regards, Email Provider"

(Continued)

Table 5 (continued)

Category	Example Prompt Structure	Example Generated Email
Non-Phishing	"Write a legitimate email from IT reminding them about the password update. Here are two example emails: [Subject: Recent Login to Your Account Hello, We noticed a login to your account from a new device. If this were you, no further action is needed. If you don't recognize this activity, please contact our support team. Thanks, Security Team], [Subject: Password Expiry Notification Hi [User's Name], This is a reminder that your current password will expire in 3 days. Please update it at your earliest convenience through the official company portal. Best, IT Department] — Now write a similar realistic email: Subject:"	"Subject: Reminder for Updating Your Password Body: Dear User, You are receiving this message as we have detected an outdated or weak password associated with your account. It is essential to keep your account secure and protect against potential threats. We recommend updating your password immediately by visiting our password policy page on our website (www.websiteurl.com/policy). Please follow the guidelines provided there regarding length, complexity, and including a combination of uppercase letters, lowercase letters, numbers, and special characters. To help ensure continued security, consider enabling multi-factor authentication on your accounts if not already enabled. This additional layer of protection provides an added measure of safety when accessing sensitive information online. Thank you for taking these steps to improve the overall security of your account and prevent unauthorized access. Should you encounter any difficulties during this process, feel free to reach out to our technical support team via email (third) or live chat service available on our website. Stay safe! Warm Regards, Carlson and Sons Crane Group"

3.2.2 Fine-Tuning the LLM

This approach involved the following steps:

- Dataset Preparation: The training dataset was prepared using a chat-based prompt completion schema. Each email sample was turned into a short dialog with three parts: a system message that sets the assistant's role, a user message containing the actual email text, and an assistant message, which was either "phishing" or "non-phishing", depending on the label. This format, which is shown in Fig. 3, helped the model understand the structure of the task and made the training more consistent with how we will use the model later. Two key steps ensured the model was trained effectively: data balancing and training–validation split. First, the dataset underwent data balancing, where the phishing and non-phishing classes were adjusted to an equal representation by under-sampling the larger class to match the minority class and thus prevent class imbalance. Subsequently, the dataset was divided into training and validation sets.
- Fine-Tuning the Model: To generate realistic emails for detecting phishing attacks in different situations, we applied a fine-tuning step to the DeepSeek-LLM-7B-Chat model [30]. We fine-tuned two separate models: one for phishing emails and one for non-phishing emails. This allowed each model to focus only on its specific type of data, which helped improve accuracy and made the synthetic email generation process more targeted and realistic. Instead of feeding plain emails directly into the model, we used

a formatted dataset. For the fine-tuning process, we used LoRA (Low-Rank Adaptation) [37], which makes training more efficient by updating only a small portion of the model's parameters. This saves memory and speeds up training without affecting performance. The key training parameters were as follows:

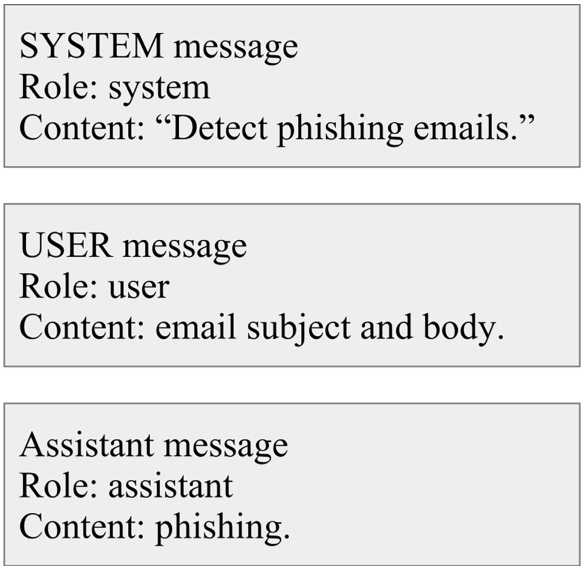

Figure 3: Chat-based prompt formatting of labeled email data.

Learning rate = 2e−5, a safe and commonly used rate for similar tasks.

Max sequence length = 256 tokens, enough to handle most email lengths while keeping memory usage low.

Batch size = 1, with gradient accumulation steps = 4 to simulate a batch size of 4 without needing more memory.

Epochs = 1, to avoid a long training time and reduce the risk of overfitting.

These parameters were chosen based on various factors, including commonly recommended values [38]. For example, the learning rate is well-suited for fine-tuning LLMs. We experimented with multiple batch sizes before deciding to use a small batch size combined with gradient accumulation. This approach balances memory constraints and training stability. Regarding the number of epochs, since our dataset is relatively large and balanced, and the model tends to overfit quickly on classification tasks, we limited training to a single epoch to reduce training time and prevent overfitting. During training, we also included a validation step to check if the model was learning the task properly and not just memorizing the data. After completing the fine-tuning, we used the trained models to start generating synthetic emails for both categories.

- Synthetic Dataset Generation: We implemented a custom training loop tailored for text generation tasks, providing greater control over how inputs and outputs were processed. This approach ensured that the model consistently learned to generate accurate responses labeled as either "phishing" or "non-phishing." By directly managing the training dynamics, we were able to guide the model's learning process more effectively than with standard training routines. To make the models generate the desired emails, we used the same parameters and scenarios as in the prompt engineering process. For the phishing email dataset, each scenario represented a common phishing tactic, such as impersonating a bank or shipping service. For the non-phishing email dataset, we gave the model instructions to

create legitimate emails that are close to real, legitimate emails. These prompts were carefully written to guide the models into generating contextually appropriate and valid emails. Similarly to the prompt engineering process, twenty thousand emails were generated.

After generating around 40,000 synthetic emails (prompt and fine-tuning), the new data followed the same structure as the cleaned version of the original dataset. Even though the original dataset contains about 80,000 emails, it still lacks enough variety in phishing samples to cover the different techniques attackers use. Synthetic data help fill this gap by adding new examples of phishing emails, improving the suitability of the dataset for handling how phishing tactics keep changing over time. Fig. 4 overviews the synthetic data generation processes.

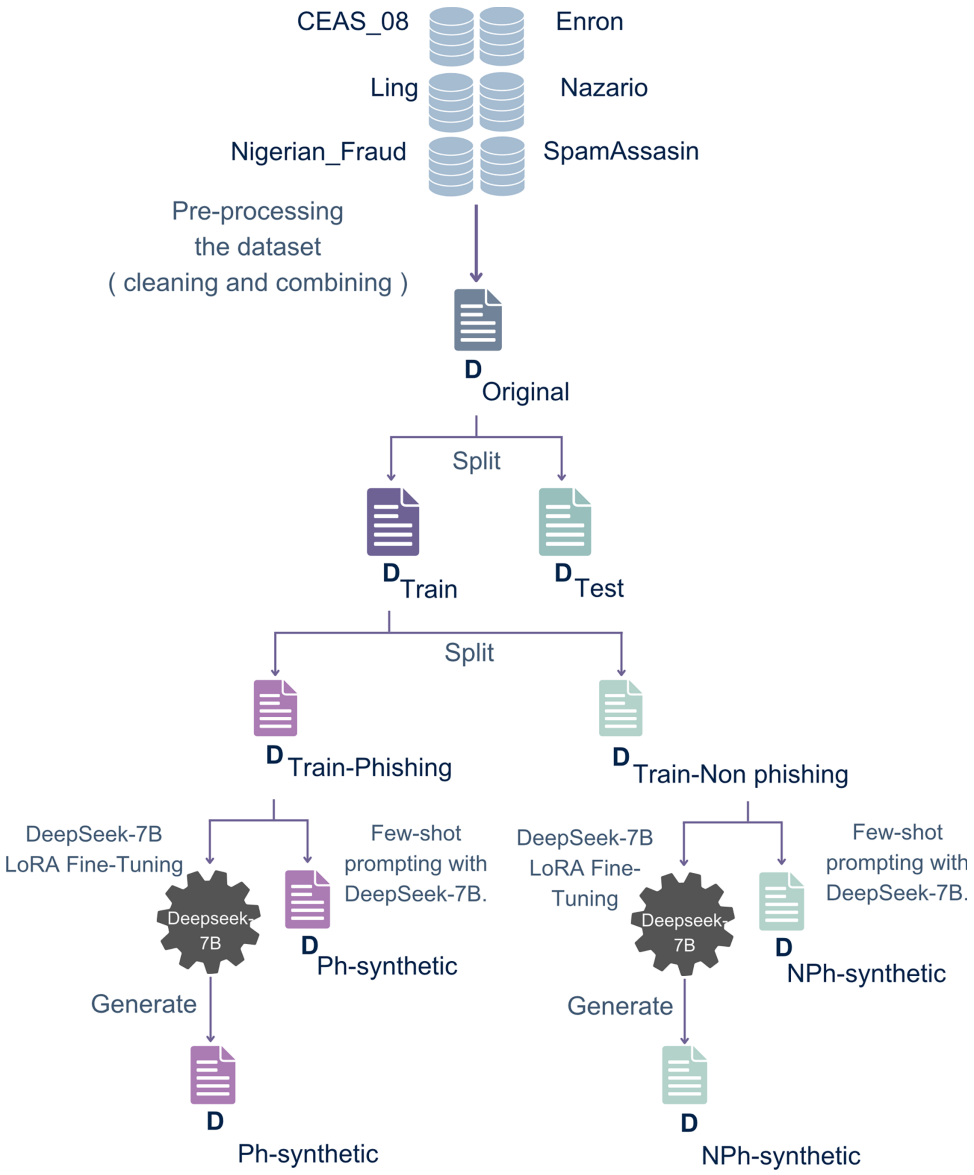

Figure 4: Synthetic data generation processes.

3.3 Evaluation of Synthetic Datasets

We evaluated two synthetic datasets, D_{Synth0} and D_{Synth1}, generated through prompt engineering for the LLM and for fine-tuning LLMs. The evaluation involved assessing data quality using multiple complementary metrics to provide a comprehensive evaluation of their effectiveness. The synthetic dataset was assessed using semantic similarity, diversity, and fluency to ensure quality and variation:

1. Semantic Quality: The contextual similarity with the original dataset was evaluated using BERTScore [39]. We divided the original dataset into the same predefined scenarios used for generations, assigning each sample to a scenario based on keyword matching. BERTScore is an evaluation metric used to evaluate the performance of BERT-based models [39], similar to the F1-score. Instead of relying on exact token matches, it measures semantic similarity by comparing the embeddings of words in predicted and reference sentences.

$$Precision = \frac{1}{|y|}\sum_{y_j \in y} max_{x_i \in x} cosine\_similarity(E(x_i), E(y_j)) \tag{1}$$

$$Recall = \frac{1}{|x|}\sum_{x_i \in x} max_{y_j \in y} cosine\_similarity(E(x_i), E(y_j)) \tag{2}$$

$$\text{BERTScore} = \text{F1}(Precision, Recall) \tag{3}$$

where x = reference, y = predicted sentences, E (x_i) = the embedding of the i-th token in x, and F1 = the harmonic mean of precision and recall.

2. Diversity: This metric was assessed using Self-Bilingual Evaluation Understudy (SelfBLEU) [40], which evaluates variation among synthetic samples. Self-BLEU is a metric used to evaluate the diversity of a dataset It calculates the Bilingual Evaluation Understudy (BLEU) score [41], which is a metric that evaluates text generation quality by measuring n-gram overlap between generated outputs and reference texts, reflecting how closely they match in content and phrasing. It was calculated by comparing each generated sample against the rest of the generated samples as the reference set. A higher Self-BLEU score indicates less diversity.

$$\text{Self\_BLEU}_i = BLEU(G_i, \{G_1, G_2, \ldots.., G_k\} \setminus \{G_i\}) \tag{4}$$

$$\text{Self\_BLEU} = \frac{1}{k}\sum_{i=1}^{k} \text{Self\_BLEU}_i \tag{5}$$

where $\{G1, G2, \ldots, G_K\}$ = the set of generated texts, and Self-BLEU$_i$ = the score for example G_i.

3. Fluency: This metric is evaluated based on perplexity [42] and calculated using a pre-trained GPT-2 language model to evaluate text naturalness. Perplexity is a metric used to evaluate language models [42]. It measures how well a probability distribution predicts a sample. Lower perplexity indicates better predictive power.

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} logP(w_i \mid w_1, w_2, \ldots.w_{i-1})\right) \tag{6}$$

where P ($w_i \mid w_1, w_2, \ldots, w_{i-1}$) = the probability of the i-th word given the preceding words, and N is the total number of words in the sequence.

Together, these evaluation metrics allow us to assess how well each approach can generate synthetic data that are semantically aligned with the original dataset while maintaining diversity and linguistic naturalness.

3.4 Classifier Training and Evaluation

To evaluate the effectiveness of synthetic datasets in training classification models, we used DistilBERT classifiers across various training setups. DistilBERT is a compact and efficient version of BERT, developed with Hugging Face [43]. The classifier training and evaluation process was as follows:

Dataset Splits: For training and testing dataset ratios, previous research has demonstrated that both 80/20 and 70/30 splits can be effective; however, the choice may depend on the size and role of each dataset, in addition to the specific machine learning task [44,45]. In our case, different dataset split strategies were applied depending on the dataset size. For example, $D_{OrigTrain}$ was divided into 70% for training, 15% for validation, and 15% for testing. The synthetic dataset D_{Synth} was smaller and thus divided into 80% for training, 10% for validation, and 10% for testing. For the synthetic data, a 70/15/15 split would have resulted in too few training samples for effective learning. This adjustment ensured sufficient training data while maintaining reliable validation and test sets.

Dataset Preparation: We evaluated the quality of two types of synthetic data generated via prompt engineering and model fine-tuning. We used three dataset configurations: $D_{OrigTrain}$—original dataset only; D_{Synth}—synthetic dataset only; and $D_{Synth+OrigTrain}$—a combination of the original and synthetic data.

Model Training: Three DistilBERT classifiers were trained on the above dataset configurations: model 1 on $D_{OrigTrain}$, model 2 on D_{Synth}, model 3 on $D_{Synth+OrigTrain}$. Then, prompt-generated and fine-tuned synthetic datasets were evaluated independently to determine their respective impact on model performance. All models were fine-tuned using parameter-efficient training with LoRA applied to the attention layers of the transformer. This limited the number of trainable parameters to approximately 1.5% of the full model.

Training Configuration: To ensure consistency across experiments, the following hyperparameters were held constant:

- Learning Rate: 5e−5;
- Batch Size: 16;
- Epochs: 3;
- Optimizer: AdamW (a variant of Adam that decouples weight decay).

The DistilBERT hyperparameters were selected based on recommendations reported in the literature. Prior studies indicate that moderate to low learning rates are effective in achieving higher classification accuracy [46]. Additionally, empirical evidence suggests that smaller batch sizes can improve model performance. With respect to optimization, Adam and its variants are widely recommended due to their robustness and effectiveness [47]. This evaluation framework supports a structured analysis of how different training data compositions influence classification performance and model generalization across original and synthetic domains.

Fig. 5 illustrates the evaluation processes.

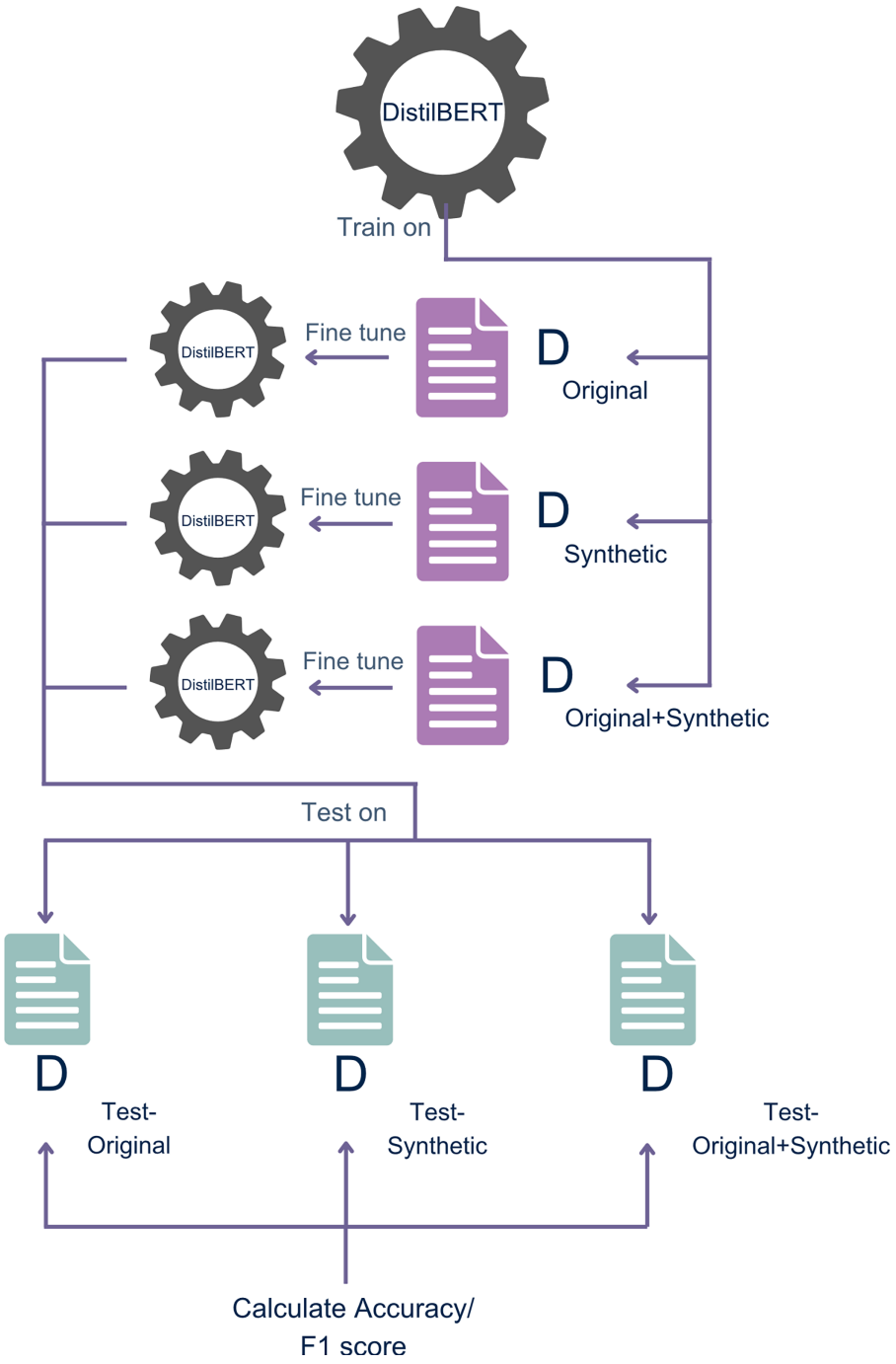

Figure 5: Evaluation process.

4 Results

4.1 Experimental Settings

The proposed solutions were implemented using two primary development platforms: Jupyter Notebook [48] for local development and Kaggle [49] for cloud-based training and text generation. The former was primarily used for dataset preprocessing and synthetic dataset generation, whereas Kaggle's

hosted environment was leveraged for fine-tuning, synthetic dataset generation, and overall evaluation. The programming language used was Python, chosen for its libraries and frameworks that support both CPU and GPU computation. PyTorch [50] was employed for model fine-tuning, while the Transformers library from Hugging Face [33] enabled efficient model handling. Support tools such as NumPy, Pandas, and Scikit-learn were utilized for data manipulation and preprocessing. The DistilBERT experiments were implemented using the same Python-based tool chain: PyTorch and the Transformers library facilitated model access and training, while data preprocessing was performed using NumPy, Pandas, and Scikit-learn. To evaluate the performance of the phishing email classification models, we employed commonly used evaluation metrics, including accuracy, precision, recall, and F1-score.

4.2 Experimental Results

In this section, we present the outcomes of our experimental evaluation, which aimed to determine the effectiveness of synthetic data in enhancing phishing email detection. The synthetic dataset was produced using prompt engineering and fine-tuning of the DeepSeek-7B-Chat model [30], with the goal of addressing limitations related to limited data availability and class imbalance. We assessed both the quality of the generated synthetic emails and performance of the DistilBERT classification models.

4.2.1 Synthetic Dataset Evaluation

The quality of the synthetic datasets was evaluated using BERTScore [39], Self-BLEU [41], and perplexity [42]. To ensure a fair comparison using BERTScore, the $D_{OrigTrain}$ dataset was partitioned into the same scenario categories used in generating D_{Synth}, which resulted in 12,000 samples. For each scenario, we computed BERTScore F1 by comparing the synthetic emails with the original emails within that scenario. We also calculated the number of phishing and non-phishing samples. Figs. 6 and 7 show the BERTScore F1-score of the models (prompt-based and fine-tuned) on phishing and non-phishing scenarios, respectively.

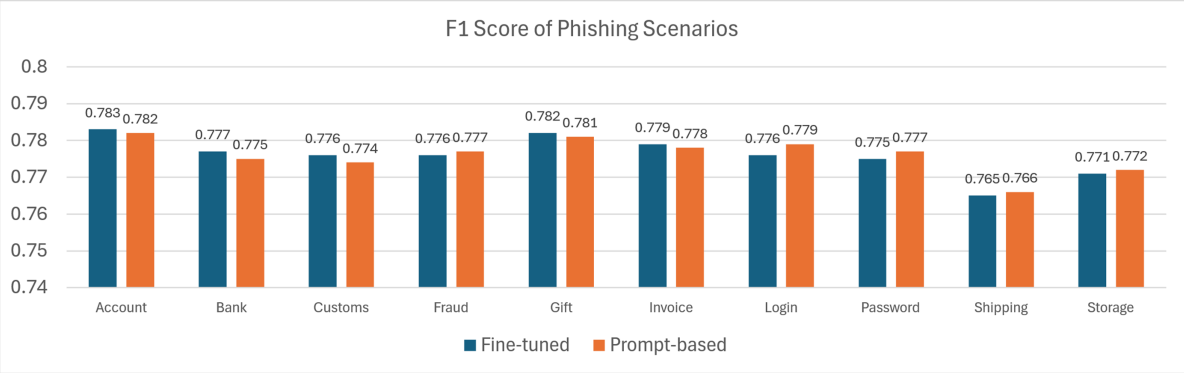

Figure 6: BERTScore F1-score of models on phishing scenarios.

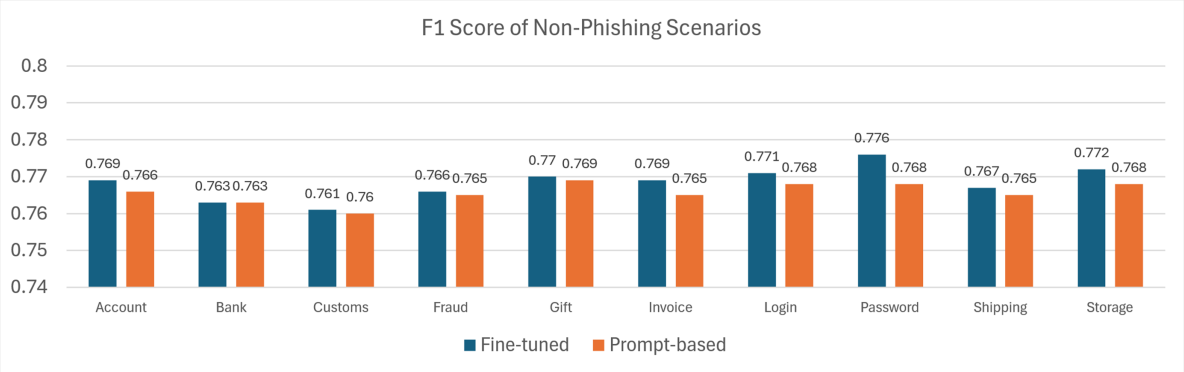

Figure 7: BERTScore F1-score of models on non-phishing scenarios.

As shown in Figs. 6 and 7, the dataset generated with the fine-tuned model achieved slightly higher scores than the prompt-based model. However, the differences were minimal—generally not exceeding 0.008 points across most scenarios.

Fig. 8 presents average Self-BLEU scores across all four synthetic email categories (fine-tuned/prompted and phishing/non-phishing). Analysis reveals that fine-tuned approaches consistently produced more diverse content, with fine-tuned non-phishing emails demonstrating the lowest Self-BLEU scores and therefore the greatest linguistic variation. In contrast, prompt-based methods yielded significantly higher Self-BLEU scores, indicating more repetitive language patterns across generated samples. Between the two prompt categories, phishing emails showed marginally better performance with slightly lower Self-BLEU scores than their non-phishing counterparts, suggesting somewhat reduced text redundancy. These findings further support the overall superiority of fine-tuned approaches in generating varied and natural-sounding content. An important hyperparameter in the Self-BLEU calculation is the n-gram size. In this paper, Self-BLEU was computed using 4-g (BLEU-4), a widely adopted standard in text generation evaluation. It captures both local and broader phrase patterns, offering a balanced view of linguistic diversity. Using fewer n-grams (e.g., 1 or 2) may miss contextual structure, while higher values often result in sparse matches. BLEU-4 provides a practical trade-off between detail and reliability, making it well-suited for this analysis.

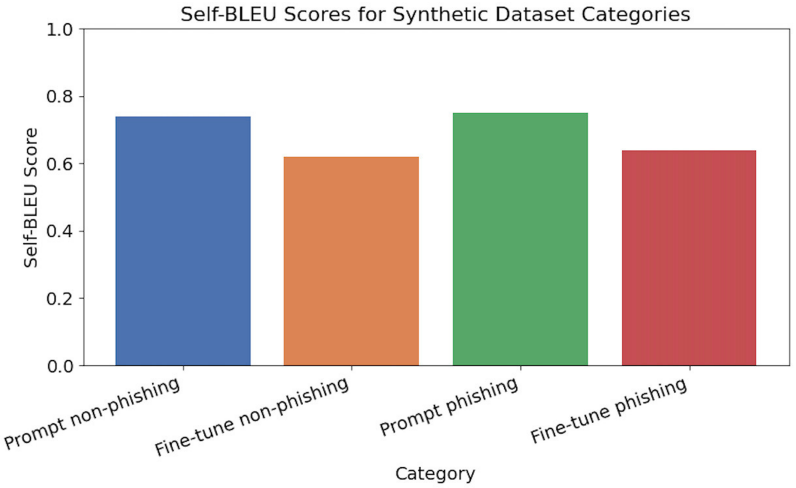

Figure 8: Average Self-BLEU score across all four synthetic email categories.

Similarly to for the Self-BLEU score, the four synthetic datasets were tested on perplexity. Figs. 9 and 10 illustrate the perplexity distributions of synthetic phishing and non-phishing emails generated using prompt-based and fine-tuned approaches, respectively. The fine-tuned model produces generally lower and more concentrated perplexity values for both classes, indicating more stable language generation behavior. In contrast, the prompt-based approach shows notable variability, particularly for phishing emails, where the distribution exhibits a wider spread and a longer tail toward higher perplexity values. Non-phishing emails generated using the prompt-based approach are more narrowly concentrated at lower perplexity levels. Overall, when comparing the two approaches, the fine-tuned method demonstrates more consistent and balanced perplexity distributions across phishing and non-phishing emails.

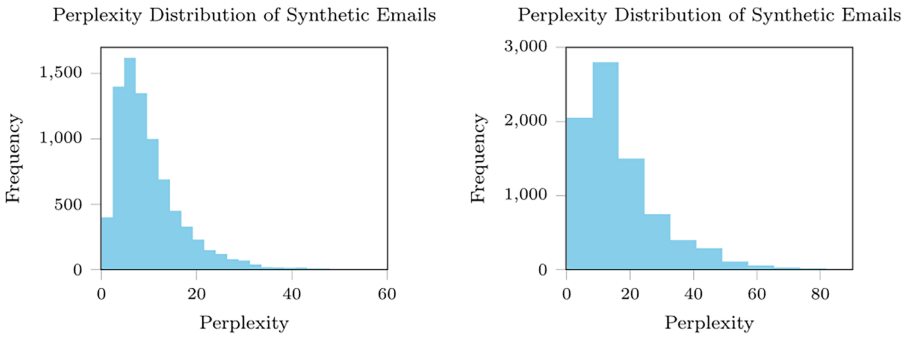

Figure 9: Perplexity distribution of phishing (**right**) and non-phishing (**left**) emails generated with prompt-based models.

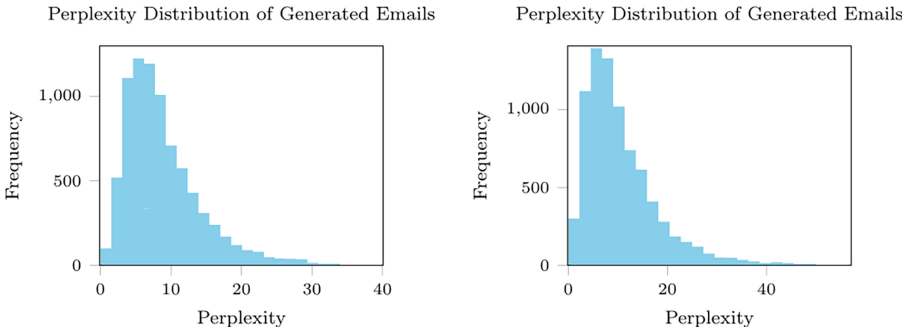

Figure 10: Perplexity distribution of phishing (**right**) and non-phishing (**left**) emails generated by fine-tuned models.

4.2.2 DistilBERT Model Performance

The evaluation results of different DistilBERT models are presented in Table 6, while Figs. 11 and 12 visualize the accuracy and F1-score, respectively. The models were trained on various datasets—original ($D_{OrigTrain}$), prompt-based synthetic (Prompt-Based D_{Synth}), fine-tuned synthetic (Fine-Tune based D_{Synth}), and their combinations ($D_{Synth+OrigTrain}$)—and subsequently tested on the independent original and synthetic datasets. The results show that models trained on $D_{OrigTrain}$ generally show high accuracy/F1-scores on the original test set (0.96/0.96), on Combined Prompt-Based $D_{Synth+OrigTrain}$ (0.89/0.89), and on Fine-Tune based $D_{Synth+OrigTrain}$ (0.91/0.91). However, their performance drops significantly when tested on prompt-based D_{Synth} (0.46/0.44). Additionally, models trained on synthetic data (Prompt-Based D_{Synth} and Fine-Tune-based D_{Synth}) achieve better performance on synthetic test sets but lower performance on the original test set. Also, combining the original and synthetic training data (Combined Prompt-Based

$D_{Synth+OrigTrain}$ and Combined Fine-Tune based $D_{Synth+OrigTrain}$) provides a good balance, maintaining high performance on the original test set while improving results on the synthetic test sets.

Table 6: Evaluation results of different DistilBERT models (accuracy/F1-score).

Training Configuration	Original Test Set	Prompt-Based Synthetic Test Set	Fine-Tuned Synthetic Test Set	Original + Prompt Synthetic Test Set	Original + Fine-Tuned Synthetic Test
$D_{OrigTrain}$	0.96/0.96	0.46/0.44	0.61/0.60	0.89/0.89	0.91/0.91
Prompt-Based D_{Synth}	0.48/0.47	0.86/0.86	–	0.53/0.52	–
Fine-Tune based D_{Synth}	0.64/0.64	–	0.81/0.81	–	0.66/0.66
Combined Prompt-Based $D_{Synth+OrigTrain}$	0.95/0.94	0.78/0.78	–	0.93/0.93	–
Combined Fine-Tune based $D_{Synth+OrigTrain}$	0.95/0.95	–	0.77/0.77	–	0.92/0.92

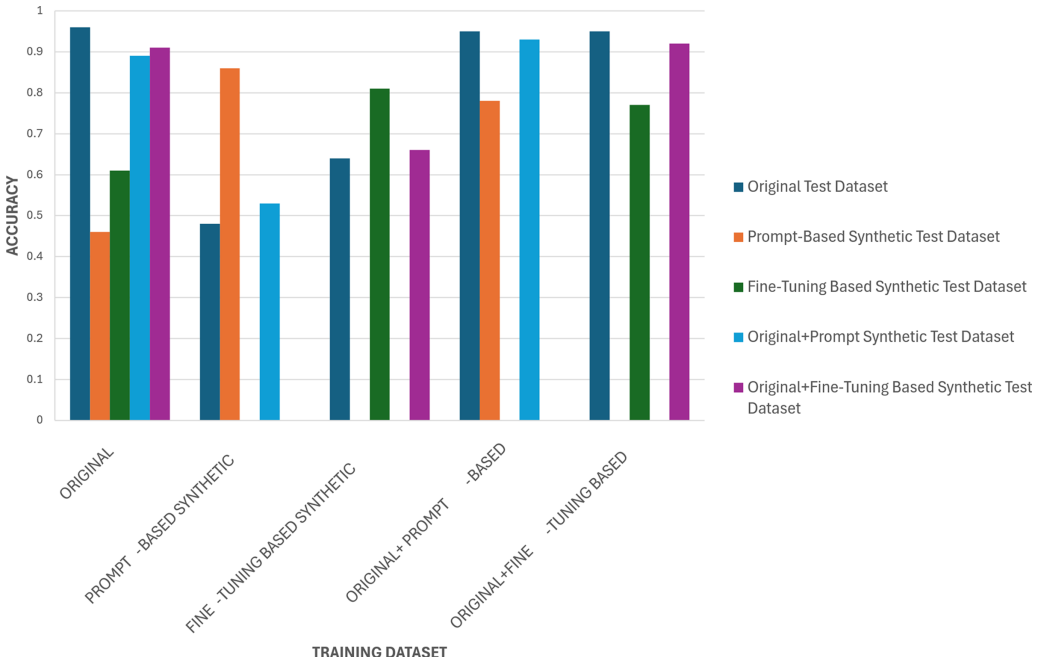

Figure 11: Accuracy of DistilBERT models.

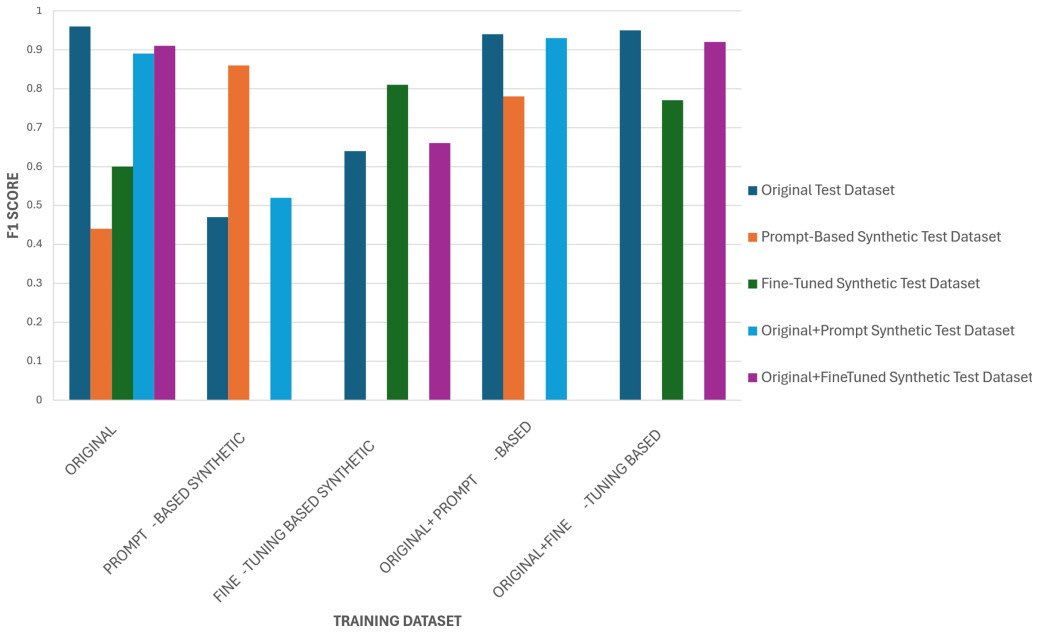

Figure 12: F1-score of DistilBERT models.

To gain deeper insight into the model's performance, a targeted review of misclassified examples was conducted, aiming to identify patterns or features that may have contributed to incorrect predictions.

False Positives: Some safe emails were incorrectly marked as phishing. This usually happened when the email used urgent or persuasive words like "act now," "limited time," or "urgent," which made them look like phishing attempts. The exact numbers of samples drawn from each test set is provided in Table 7.

Table 7: False positives—sample distribution across test sets.

Training Configuration	Original Test Set	Prompt-Based Synthetic Test Set	Fine-Tuned Synthetic Test Set	Original + Prompt Synthetic Test Set	Original + Fine-Tuned Synthetic Test
$D_{OrigTrain}$	305	642	537	947	842
Prompt-Based D_{Synth}	3899	149	–	4048	–
Fine-Tune based D_{Synth}	1403	–	164	–	1567
Combined Prompt-Based $D_{Synth+OrigTrain}$	304	191	–	495	–
Combined Fine-Tune based $D_{Synth+OrigTrain}$	315	–	182	–	497

False Negatives: Some phishing emails were not detected, especially those that looked like normal internal company messages. These were often carefully written and hard to spot. The exact number of samples drawn from each test set is provided in Table 8.

Table 8: False negatives—sample distribution across test sets.

Training Configuration	Original Test Set	Prompt-Based Synthetic Test Set	Fine-Tuned Synthetic Test Set	Original + Prompt Synthetic Test Set	Original + Fine-Tuned Synthetic Test
$D_{OrigTrain}$	246	317	240	563	486
Prompt-Based D_{Synth}	2506	99	–	2605	–
Fine-Tune based D_{Synth}	3035	–	225	–	3260
Combined Prompt-Based $D_{Synth+OrigTrain}$	324	199	–	523	–
Combined Fine-Tune based $D_{Synth+OrigTrain}$	313	–	281	–	594

The model trained on original data showed a small difference between how well it handled phishing and non-phishing emails, as presented in Tables 7 and 8. The numbers of false positives (305) and false negatives (246) are close, which indicates that the model performs similarly in both classes. The model trained on original data was slightly better at detecting phishing emails, as shown by its lower number of false negatives. The inclusion of synthetic data, particularly from the fine-tuned DeepSeek-LLM-7B-Chat, improved F1-scores, narrowing the performance gap between classes. These results demonstrate the role of synthetic data in enhancing the model's ability to recognize phishing attempts.

The inclusion of synthetic data resulted in noticeable improvements in classification accuracy and F1-score:

- Combined Prompt-Based Approach: The Combined Prompt-Based $D_{Synth+OrigTrain}$ model improved the performance on the original test set from 0.48 (Prompt-Based D_{Synth} alone) to 0.95. On the original + prompt synthetic test set, it achieved 0.93, showing significant enhancement over the individual approaches.
- Combined Fine-Tune Based Approach: The Combined Fine-Tune based D_{Synth}+$_{OrigTrain}$ model improved on the original test set from 0.64 (Fine-Tune-based D_{Synth} alone) to 0.95. On the original + fine-tuned synthetic test set, it reached 0.92.
- Cross-Dataset Performance: The $D_{OrigTrain}$ model scored 0.96 on the original test set but only 0.46 and 0.61 on the prompt-based and fine-tuned test sets, respectively. The Prompt-Based D_{Synth} model scored 0.48 on the original test set and 0.86 on its matching test set. The Fine-Tune based D_{Synth} model scored 0.64 on the original test set and 0.81 on its matching test set.
- Best Overall Performance: The combined approach models showed F1-scores of 0.93 and 0.92 on combined test sets. The Original DOrigTrain model achieved 0.89 and 0.91 on the combined test sets. Table 6 shows that models trained on original and synthetic training data consistently outperformed single-source trained models, particularly when evaluated against diverse test sets that include both original and synthetic phishing examples. Models trained on the combined datasets showed stronger generalization capabilities: when tested on unseen synthetic data, the combined models maintained relatively high accuracy and F1-scores, compared to models trained on original data alone. This suggests that synthetic examples exposed the model to a wider variety of linguistic structures and phishing tactics.

5 Discussion

The performance improvement observed when integrating synthetic data can be attributed to several factors, directly linked to the quality of the synthetic samples generated using prompt engineering and

the fine-tuning of the DeepSeek-LLM-7B-Chat model. The fine-tuned model's synthetic emails consistently achieved higher BERTScore values due to its training approach. Also, the fine-tuned model learned directly from real email data, allowing it to replicate authentic language patterns, maintain semantic coherence, and capture nuanced details in text. In contrast, the prompt-based approach used pre-written instructions without directly learning from data. This limited its ability to generate emails that semantically resembled real examples, leading to slightly lower BERTScore values. Fine-tuned synthetic samples demonstrated lower Self-BLEU scores, especially in non-phishing categories, because the model was trained to produce diverse, contextually appropriate emails rather than repeating patterns. Fine-tuning exposed the model to varied training examples, teaching it to create datasets with diversity. The prompt-based approach, however, generated more repetitive emails because it relied on predefined templates, making it less capable of creating diverse content. Fine-tuned emails consistently achieved lower perplexity values, reflecting better fluency and grammatical accuracy because the fine-tuned model directly learned sentence structure, phrasing, and coherent language use from real data. Conversely, the prompt-based approach used static instructions, leading to inconsistent quality. In phishing emails, where creative variations are critical, the prompt-based method struggled, producing texts with higher perplexity. The fine-tuned model's performance can be directly attributed to its training on real data, allowing it to generate contextually accurate and diverse content. In contrast, the prompt-based method, while faster and simpler, lacked the adaptive learning capabilities needed for high-quality email generation.

Moreover, the results confirm our hypothesis that generative AI can be an effective solution for dataset limitations in phishing detection. By fine-tuning the DeepSeek-LLM-7B-Chat model [30], we were able to generate realistic and structurally coherent phishing and non-phishing emails. A performance mismatch is observed when models are trained solely on synthetic data and evaluated on original emails. This behavior is expected and aligns with observations in the literature [51], where synthetic-only classifiers have been shown to significantly underperform on sensitive datasets (e.g., hate speech and toxic language datasets). These findings indicate that synthetic augmentation may be more effective for subtle or less explicitly harmful content. It is important to emphasize that this particular setting (training on synthetic data and testing on original data) is not intended to demonstrate that synthetic data can fully replace real data, but rather to examine its value as an additional training signal. When combined with the original dataset, these synthetic samples significantly enhanced the performance of the DistilBERT classifier—especially in the F1-score for the phishing class. This finding is in line with prior research emphasizing the effectiveness of combining synthetic and real data. For instance, Kang et al. [52] showed that combining synthetic and original datasets significantly improved model performance across multiple benchmarks, especially in settings with data scarcity. Similarly, the study by Whitfield [53] highlighted that the most significant improvement was observed when synthetic data was used as augmentation rather than replacement, reinforcing our approach. When training on the combined dataset, the model maintains strong performance on real data (F1 = 0.94) while also improving performance on synthetic data (F1 = 0.78). This behavior is expected due to the distribution of the two datasets (original vs. synthetic). The original dataset constitutes the dominant portion of the training data, whereas synthetic emails represent a smaller portion. As a result, synthetic data primarily act as regularization samples that improve semantic coverage and robustness, rather than serving as a target domain for optimization. In our study, these results show that synthetic data enhances model robustness without degrading real-data performance. The improvement observed with the combined dataset is therefore not due to increased data size alone, but rather to increased data diversity and exposure to varied phishing styles. These findings clarify that synthetic data is most effective as an augmentation strategy rather than as a standalone training source.

DeepSeek-LLM-7B-Chat [30] showed strong ability in generating realistic emails. After applying LoRA fine-tuning and quantization, the model produced emails that were clear, varied, and close in style to real phishing and non-phishing emails. This was confirmed by good BERTScore, Self-BLEU, and perplexity scores. As an open source model designed for researchers, DeepSeek-LLM-7B-Chat was easier to control and customize than closed source, commercial LLMs. It provides greater flexibility during training and generation, enabling the creation of more realistic and diverse examples, including emails that incorporate tricky or subtle language.

To place our findings in the context of the existing literature, it is important to note that the high accuracy values reported in previous phishing detection studies are typically obtained under non-comparable experimental conditions. Many prior works rely on datasets that are limited in size or have imbalanced class distributions. Consequently, the primary motivation of this work is to address a fundamental limitation in phishing detection research: the lack of diverse and representative training datasets. In this study, we construct a more comprehensive dataset by integrating six publicly available phishing datasets with newly generated synthetic emails produced using DeepSeek-7B-Chat. This dataset better captures the diversity and evolving nature of modern phishing tactics. Accordingly, our evaluation emphasizes generalization performance rather than overfitting to a limited dataset. Our results, therefore, highlight robustness and practical relevance, rather than perfect performance on an individual dataset.

Our findings have direct practical value for organizations with limited phishing data; they can use generative models to augment datasets, enabling better detection performance without extensive manual data collection. Also, our approach is particularly useful in low-resource settings or for new/emerging phishing tactics that are underrepresented in existing datasets. While the proposed approach showed strong results, several limitations were noted. For example, some generated emails were too generic, potentially limiting diversity. Thus, it would be worth investigating whether using real emails as few-shot seeds, rather than strictly structured crafted examples, would improve the authenticity and diversity of the generated emails. Also, the current study focused only on English-language emails. The model did not explore advanced phishing variants, such as multi-language, visual, or spear-phishing attacks. Future research could focus on fine-tuning larger LLMs such as GPT-4, Claude, or DeepSeek-LLM-67B, and comparing their performance to DeepSeek-LLM-7B-Chat. Additionally, expanding synthetic data generation to include non-English or multimodal phishing attacks (e.g., emails with malicious attachments or images). In the context of phishing emails, it is important to combine traditional NLP metrics with more specialized measures, such as phishing realism, persuasiveness, social engineering features, or deception quality. However, for this study, this would require human evaluation and may not be easily automated. Nevertheless, this represents a promising direction for future work and is worth further investigation.

6 Conclusions

Phishing continues to be a serious cybersecurity problem, with tricks used to deceive people and steal their personal information. As attackers start using AI to write more convincing phishing emails, it is becoming harder for traditional systems to detect them. Generative AI has strong potential, not only in simulating phishing attacks but also in helping improve detection through the creation of high-quality phishing emails to train detection models. In this study, we used the Deepseek-7B-Chat model to create realistic phishing and non-phishing emails and tested two methods: prompt engineering and fine-tuning. We evaluated the generated emails using BERTScore, Self-BLEU, and perplexity. The results show that using both approaches generates more realistic and diverse emails. We tested three data set setups: one with original emails, one with only synthetic emails, and one that combined both. We used these datasets to train DistilBERT classifiers and evaluate them with different types of test data. The results show that the combined

datasets achieved the strongest results, indicating that incorporating synthetic data improves the model's robustness and overall reliability.

Overall, this study shows that using generative AI models like DeepSeek for synthetic dataset generation contributes to developing more robust, adaptive, and intelligent phishing detection systems that can effectively counteract evolving phishing tactics. Additionally, generative AI can significantly strengthen cybersecurity defenses and reduce the overall success rate of phishing attacks. It can also help solve a major problem in phishing research: the lack of high-quality, diverse labeled emails. In the future, we plan to expand the generation of synthetic phishing emails to cover a broader range of scenarios, including multilingual content. We aim to continue refining prompt engineering techniques and develop more fine-tuned models capable of capturing diverse phishing styles. Additionally, we intend to evaluate the robustness of these systems through real-world attack simulations.

Acknowledgement: Not applicable.

Funding Statement: This work has been funded by the Onging Research Funding program (ORF-2026-857), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: The authors confirm their contributions to this paper as follows: conceptualization: Isra Al-Turaiki, Amani Al-Ajlan and Najwa Altwaijry; methodology: Isra Al-Turaiki, Lama Almelaifi, Remas Alharbi, Shahad Al-Hussain and Fay Alfarraj; software, validation, resources, data curation, and visualization: Lama Almelaifi, Remas Alharbi, Shahad Al-Hussain and Fay Alfarraj; formal analysis: all authors; writing—original draft preparation: Amani Al-Ajlan; writing—review and editing: all authors; supervision: Isra Al-Turaiki and Najwa Altwaijry; project administration: Isra Al-Turaiki; funding acquisition: Najwa Altwaijry. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available from Kaggle at https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset, and the synthetic data at https://github.com/israksu/PhishingDataset.

Ethics Approval: Not applicable.

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

References

1. APWG. Phishing activity trends reports. APWG [cited 2025 May 17]. Available from: https://apwg.org/trendsreports.
2. Cost of a data breach report 2025. IBM, 2025 [cited 2025 Dec 30]. Available from: https://www.ibm.com/reports/data-breach.
3. Altwaijry N, Al-Turaiki I, Alotaibi R, Alakeel F. Advancing phishing email detection: a comparative study of deep learning models. Sensors. 2024;24(7):2077. doi:10.3390/s24072077.
4. Eze CS, Shamir L. Analysis and prevention of AI-based phishing email attacks. Electronics. 2024;13(10):1839. doi:10.3390/electronics13101839.
5. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.
6. Das R. Generative AI: phishing and cybersecurity metrics. Boca Raton, FL, USA: CRC Press; 2024.
7. Schmitt M, Flechais I. Digital deception: generative artificial intelligence in social engineering and phishing. Artif Intell Rev. 2024;57(12):324. doi:10.1007/s10462-024-10973-2.
8. Neha N, Prabhu V. Language models in financial fraud detection: a comparative study. In: Artificial intelligence: theory and applications. Singapore: Springer Nature; 2025. p. 45–60. doi:10.1007/978-981-96-1687-9_4.

9. Gangavarapu T, Jaidhar CD, Chanduka B. Applicability of machine learning in spam and phishing email filtering: review and approaches. Artif Intell Rev. 2020;53(7):5019–81. doi:10.1007/s10462-020-09814-9.
10. SpamAssassin Public Corpus. Apache spamassassin project. 2005 [cited 2025 May 17]. Available from: https://spamassassin.apache.org/old/publiccorpus/.
11. Nazario J. Nazario phishing email corpus. [cited 2025 Dec 25]. Available from: https://monkey.org/~jose/phishing/.
12. Valecha R, Mandaokar P, Rao HR. Phishing email detection using persuasion cues. IEEE Trans Dependable Secure Comput. 2021;19(2):747–56. doi:10.1109/tdsc.2021.3118931.
13. Phishing Scams and Spoof Emails at MillerSmiles.co.uk. [cited 2025 May 17]. Available from: http://www.millersmiles.co.uk/.
14. Klimt B, Yang Y. The enron corpus: a new dataset for email classification research. In: Machine learning: ECML 2004. Berlin/Heidelberg, Germany: Springer; 2004. p. 217–26. doi:10.1007/978-3-540-30115-8_22.
15. Alammar M, Badawi MA. Phishing email detection using machine learning techniques. Int J Comput Sci Netw Secur. 2022;22(5):277–83. doi:10.5220/0008119805290534.
16. Tan CL. Phishing dataset for machine learning: feature evaluation. Mendeley Data. 2018. doi:10.17632/h3cgnj8hft.1.
17. EmailHeaderAnomalyDetection. [cited 2025 May 17]. Available from: https://github.com/kregg34/EmailHeaderAnomalyDetection.
18. Doshi J, Parmar K, Sanghavi R, Shekokar N. A comprehensive dual-layer architecture for phishing and spam email detection. Comput Secur. 2023;133(1):103378. doi:10.1016/j.cose.2023.103378.
19. Heiding F, Schneier B, Vishwanath A, Bernstein J, Park PS. Devising and detecting phishing emails using large language models. IEEE Access. 2024;12:42131–46. doi:10.1109/access.2024.3375882.
20. Al-Subaiey A, Al-Thani M, Abdullah Alam N, Antora KF, Khandakar A, Uz Zaman SA. Novel interpretable and robust web-based AI platform for phishing email detection. Comput Electr Eng. 2024;120(7):109625. doi:10.1016/j.compeleceng.2024.109625.
21. Phishing email dataset. Kaggle. [cited 2025 Nov 13]. Available from: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset.
22. Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD. An evaluation of naive bayesian anti-spam filtering. arXiv:cs/0006013. 2000. doi:10.48550/arXiv.cs/0006013.
23. CEAS. 2008 spam challenge dataset. In: Proceedings of the Conference on Email and Anti-Spam (CEAS); 2008 Aug 21–22; Mountain View, CA, USA.
24. Radev D. CLAIR collection of fraud email (nigerian 419 scam dataset). Stroudsburg, PA, USA: ACL Data and Code Repository; 2008.
25. Atawneh S, Aljehani H. Phishing email detection model using deep learning. Electronics. 2023;12(20):4261. doi:10.3390/electronics12204261.
26. Thakur K, Ali ML, Obaidat MA, Kamruzzaman A. A systematic review on deep-learning-based phishing email detection. Electronics. 2023;12(21):4545. doi:10.3390/electronics12214545.
27. Alhuzali A, Alloqmani A, Aljabri M, Alharbi F. In-depth analysis of phishing email detection: evaluating the performance of machine learning and deep learning models across multiple datasets. Appl Sci. 2025;15(6):3396. doi:10.3390/app15063396.
28. Hosseinzadeh M, Ali U, Ali S, Abbaszadi R, Gharehchopogh FS, Khoshvaght P, et al. Improving phishing email detection performance through deep learning with adaptive optimization. Sci Rep. 2025;15(1):36724. doi:10.1038/s41598-025-20668-5.
29. Phishing emails dataset. [cited 2025 Dec 24]. Available from: https://www.kaggle.com/datasets/subhajournal/phishingemails.
30. DeepSeek-AI. Deepseek-llm-7b-Chat. Hugging Face. [cited 2025 Nov 13]. Available from: https://huggingface.co/deepseek-ai/deepseek-llm-7b-chat.
31. Verizon data breach investigations report (DBIR). Verizon business. [cited 2025 Dec 30]. Available from: https://www.verizon.com/business/resources/reports/dbir/.

32. Alsubaie S. Enhancing phishing awareness using scenario-based learning techniques [master's thesis]. Riyadh, Saudi Arabia: Saudi Digital Library; 2023.
33. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Stroudsburg, PA, USA: ACL; 2020. p. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.
34. Faker is a python package that generates fake data for you. Python. [cited 2025 May 18]. Available from: https://github.com/joke2k/faker.
35. Min S, Lyu X, Holtzman A, Artetxe M, Lewis M, Hajishirzi H, et al. Rethinking the role of demonstrations: what makes in-context learning work? arXiv:2202.12837. 2022. doi:10.48550/arXiv.2202.12837.
36. Sun H, Zhang Z, Mi F, Wang Y, Liu W, Cui J, et al. MoralDial: a framework to train and evaluate moral dialogue systems via moral discussions. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. p. 2213–30. doi:10.18653/v1/2023.acl-long.123.
37. Li Y, Song L, LoRAN Hou H. Improved low-rank adaptation by a non-linear transformation. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024; 2024 Nov 12–16; Miami, FL, USA. p. 3134–43. doi:10.18653/v1/2024.findings-emnlp.177.
38. Hugging face. Fine-tune a pretrained model. [cited 2025 May 27]. Available from: https://huggingface.co/learn/llm-course/en/chapter3/3.
39. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. Bertscore: evaluating text generation with BERT. arXiv:1904.09675. 2019. doi:10.48550/arXiv.1904.09675.
40. Zhu Y, Lu S, Zheng L, Guo J, Zhang W, Wang J, et al. Texygen: a benchmarking platform for text generation models. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval; 2018 Jul 8–12; Ann Arbor, MI, USA. p. 1097–100. doi:10.1145/3209978.3210080.
41. Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics; 2002 Jul 7–12; Philadelphia, PA, USA. p. 311–8. doi:10.3115/1073083.1073135.
42. Jelinek F, Mercer RL, Bahl LR, Baker JK. Perplexity—a measure of the difficulty of speech recognition tasks. J Acoust Soc Am. 1977;62(S1):S63. doi:10.1121/1.2016299.
43. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108. 2019. doi:10.48550/arXiv.1910.01108.
44. AlOmar MK, Khaleel F, AlSaadi AA, Hameed MM, AlSaadi MA, Al-Ansari N. The influence of data length on the performance of artificial intelligence models in predicting air pollution. Adv Meteor. 2022;2022(3):5346647. doi:10.1155/2022/5346647.
45. Bukaita W, Garcia de Celis G, Gurram M. Training-testing data ratio selection for accurate time series forecasting: a COVID-19 case study. In: Proceedings of the Future Technologies Conference (FTC) 2024. Cham, Switzerland: Springer Nature; 2024. Vol. 3, p. 227–46. doi:10.1007/978-3-031-73125-9_14.
46. Lorenzoni G, Portugal I, Alencar P, Cowan D. Exploring variability in fine-tuned models for text classification with DistilBERT. arXiv:2501.00241. 2024. doi:10.48550/arXiv.2501.00241.
47. Gkouti N, Malakasiotis P, Toumpis S, Androutsopoulos I. Should I try multiple optimizers when fine-tuning pre-trained Transformers for NLP tasks? Should I tune their hyperparameters? arXiv:2402.06948. 2024. doi:10.48550/arXiv.2402.06948.
48. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In: Positioning and power in academic publishing: players, agents and agendas. Amsterdam, The Netherlands: IOS Press; 2016. p. 87–90. doi:10.3233/978-1-61499-649-1-87.
49. Kaggle: machine learning and data science platform. Kaggle. [cited 2025 Nov 13]. Available from: https://www.kaggle.com/code.
50. PyTorch documentation—PyTorch 2.7 documentation. [cited 2025 Jan 1]. Available from: https://docs.pytorch.org/docs/stable/index.html.

51. Schmidhuber M, Kruschwitz U. LLM-based synthetic datasets: applications and limitations in toxicity detection. In: Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying @ LREC-COLING-2024; 2024 May 20–25; Torino, Italy. p. 37–51. doi:10.63317/4e9jwkpywvi7.
52. Kang A, Chen JY, Lee-Youngzie Z, Fu S. Synthetic data generation with LLM for improved depression prediction. arXiv:2411.17672. 2024. doi:10.48550/arXiv.2411.17672.
53. Whitfield D. Using gpt-2 to create synthetic data to improve the prediction performance of NLP machine learning classification models. arXiv:210410658. 2021. doi:10.48550/arXiv.2104.10658.