

# Phishing Detection in the Gen-AI Era: Quantized LLMs vs Classical Models

Jikesh Thapa, Gurrehmat Chahal, Șerban Voinea Gabreanu, Yazan Otoum

*School of Computer Science and Technology*

*Algoma University, Canada*

Emails: {jithapa, gurchahal, svoineagabreanu, otoum}@algomau.ca

**Abstract**—Phishing attacks are becoming increasingly sophisticated, underscoring the need for detection systems that strike a balance between high accuracy and computational efficiency. This paper presents a comparative evaluation of traditional Machine Learning (ML), Deep Learning (DL), and quantized small-parameter Large Language Models (LLMs) for phishing detection. Through experiments on a curated dataset, we show that while LLMs currently underperform compared to ML and DL methods in terms of raw accuracy, they exhibit strong potential for identifying subtle, context-based phishing cues. We also investigate the impact of zero-shot and few-shot prompting strategies, revealing that LLM-rephrased emails can significantly degrade the performance of both ML and LLM-based detectors. Our benchmarking highlights that models like DeepSeek R1 Distill Qwen 14B (Q8\_0) achieve competitive accuracy, above 80%, using only 17 GB of VRAM, supporting their viability for cost-efficient deployment. We further assess the models’ adversarial robustness and cost-performance tradeoffs, and demonstrate how lightweight LLMs can provide concise, interpretable explanations to support real-time decision-making. These findings position optimized LLMs as promising components in phishing defence systems and offer a path forward for integrating explainable, efficient AI into modern cybersecurity frameworks.

**Index Terms**—Phishing Attacks Detection, Large Language Models (LLMs), Cybersecurity, Deep Learning (DL), Machine Learning (ML)

## I. INTRODUCTION

Phishing is a prevalent cyberthreat that manipulates users into divulging sensitive information, such as credentials and financial data. Between November 2023 and January 2024, the Cybercrime Information Center collected approximately one million phishing reports [1]. With the increasing sophistication of phishing tactics, traditional detection approaches struggle to keep pace with evolving attack techniques. Phishing has become more challenging in the 2020s due to the increased use of social media and multi-vector attacks that combine various methods, including email, text messages, and social media platforms. They have evolved from simple, deceptive emails to highly sophisticated social engineering schemes. Attackers leverage Artificial Intelligence (AI) generated content, deepfake techniques, and multi-channel deception strategies to exploit human vulnerabilities. Integration of AI, particularly Machine Learning (ML) and Natural Language Processing (NLP), has shown promising results in alleviating some of this burden. Several recent studies published between 2021 and 2024 have demonstrated the effectiveness of deep learning models, including Convolutional Neural Networks (CNNs)

and Recurrent Neural Networks (RNNs) [2], in detecting phishing attempts with higher accuracy compared to classical approaches, with XGBoost achieving the highest accuracy at 99.89%, followed by PILFER with 99.5% [3]. However, recent advancements in Large Language Models (LLMs) and Generative AI (GenAI) [4], [5] further exacerbate the threat, enabling automated and highly targeted phishing campaigns. The authors of [6] found that while traditional phishing detectors, such as Gmail Spam Detector, SpamAssassin, Proofpoint, and State-of-the-Art LLMs perform well on original phishing emails, their accuracy and recall decline notably when dealing with LLM-rephrased versions of the duplicate emails. The study displayed a significantly better classification performance from LLMs with rephrased emails, and even with the original emails, the performance was still marginally more accurate. Unfortunately, the performance gains of LLMs come at a significant cost in terms of energy and computational resources. The total energy consumption for training a transformer model with 6B parameters to completion is estimated to be around 103.5MWh [7], and given that the computational cost of training an LLM depends on the number of parameters, modern alternatives like GPT-1, GPT-2, and GPT-3 featuring 117 million, 1.5 billion, and 175 billion parameters, respectively, the question arises whether the marginal performance gain is worth it. And if and how the LLMs or Traditional approaches can be evolved to provide a more optimal solution. This paper presents a structured review of such next-generation phishing detection techniques and their cost-to-performance ratio, focusing on the following contributions:

- A comparative analysis of traditional, Machine Learning, Deep Learning, and LLM-based phishing detection approaches.
- Exploration of adversarial phishing tactics, LLM rephrasing, and AI/LLM-driven countermeasures.
- Comparative analysis of model performances and marginal performance gain against supplemental time and resource consumption.
- Identification of research gaps and future directions in phishing detection.

The remainder of this paper is organized as follows: Section II reviews related works across ML, DL, and LLM-based phishing detection, identifying key research gaps. Section III presents the proposed models and methodologies, including experimental setup and dataset details. Section IV summarizes

the findings, highlights cost-performance trade-offs, and outlines future research directions for robust, efficient phishing detection.

## II. LITERATURE REVIEW

### A. Related Works

Phishing detection has evolved from traditional ML/NLP methods to deep learning models, and recently to LLM-driven approaches. Studies such as [8] and [9] highlight how LLMs both amplify phishing sophistication and enhance detection under adversarial conditions. Recent work also integrates LLMs into Security Awareness Training (SAT) [10] and cybersecurity policy formulation [11], reducing phishing susceptibility. The authors of [12] proposed early malicious prompt detection transferable across major LLMs, achieving high accuracy, up to 96%. Despite initial successes, conventional ML/NLP approaches [13], [14] remain inadequate against evolving phishing attacks, motivating the development of hybrid solutions that combine LLM contextual reasoning with the proven robustness of traditional methods [15].

### B. Comparative Work

1) *Adversarial Prompting: Using LLMs to Subvert Detection Models:* Using LLMs to reword emails effectively reduces the efficiency of phishing detectors, with traditional Machine Learning models losing a significant amount of accuracy. LLMs such as GPT-4, however, were shown to be able to retain more detection accuracy despite the reworded emails [6]. As shown in Fig. 1, all models achieve very high accuracy when classifying the unmodified (“original”) emails. GPT-4 tops the list at around 98.5% accuracy, while traditional models like Naive Bayes and Logistic Regression are just behind at 97.2% and 96.8%, respectively.

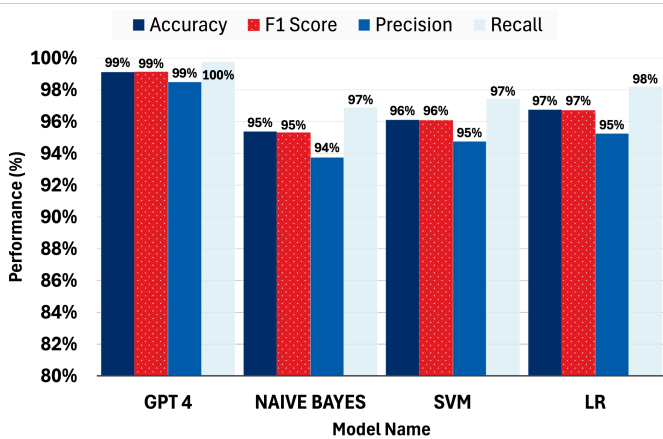


Fig. 1. Comparative results of phishing-detection models on original emails.

As illustrated in Fig. 2, zero-shot rephrasing of phishing emails leads to a noticeable decline in detection performance across all evaluated models. Specifically, Naive Bayes exhibits a drop of approximately 5.3 percentage points, Logistic Regression declines by 6.1 percentage points, and GPT-4 experiences a reduction of about 3.2 percentage points in accuracy.

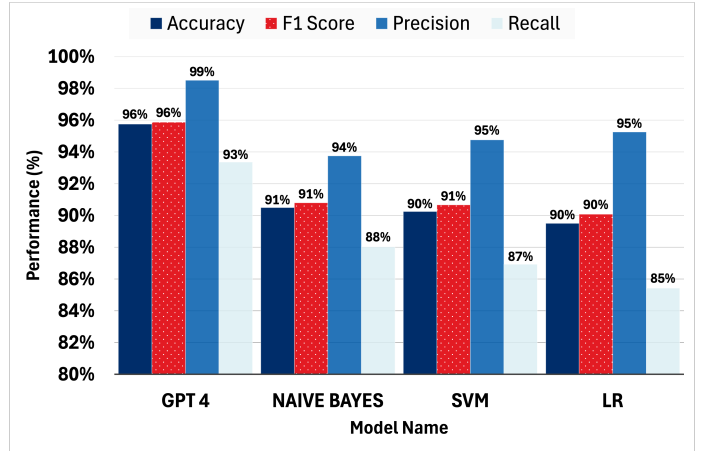


Fig. 2. Comparative results of phishing-detection models on zero-shot rephrased emails.

This performance degradation underscores the vulnerability of both traditional and LLM-based detectors to simple LLM-generated paraphrasing techniques.

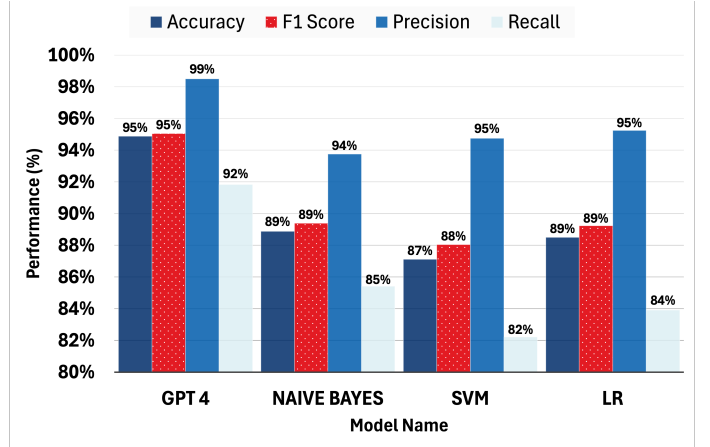


Fig. 3. Comparative results of phishing-detection models on few-shot rephrased emails.

Fig. 3 illustrates the impact of few-shot rephrasing on phishing detection performance, revealing a more pronounced decline than zero-shot rephrasing. Naive Bayes accuracy drops by 6.5 percentage points, Logistic Regression by 8.25 points, and SVM by 9.0 points. In contrast, GPT-4 demonstrates greater resilience, with a relatively modest decrease of 4.24 percentage points. These results highlight the comparative robustness of LLM-based detectors against paraphrased inputs. However, the significant degradation observed across traditional models highlights their limitations when confronted with advanced LLM-based evasion strategies, such as reflection and beam search, which effectively preserve malicious intent while obfuscating linguistic patterns.

- Debnath et al. [16] achieved 99.14% accuracy using BERT on the Enron Email Dataset
- Ali et al. [17] achieved an overall classification accuracy of 99% on the Lingspam dataset, retaining 96% accuracy

against adversarial samples

- Shahrivari et al. (2020) [18] demonstrated the effectiveness of various machine learning techniques for phishing detection

The research collectively indicates that while traditional ML approaches can be highly effective against conventional phishing attempts, they require significant enhancement to counter emerging LLM-generated threats. The defensive algorithm proposed by Fairbanks demonstrated a 97% improvement in mitigating these sophisticated attacks. This comparative analysis highlights the importance of developing specialized defenses against LLM-generated phishing, as traditional methods may be insufficient against these increasingly sophisticated threats.

2) *Traditional ML Models Performance:* The “David versus Goliath” paper by Greco et al. [19] demonstrates that smaller, traditional machine learning models can be highly effective for detecting LLM-generated phishing emails. Their experimental results show that neural networks achieved the highest accuracy at 99.78% but with the highest resource cost, closely followed by SVM (99.2%) and logistic regression (99.03%), while other models like KNNs (97.67%), Random Forest (98.16%) and Naïve Bayes (94.1%) showed relatively lower accuracy but much stronger performance. These results are further reinforced by our performance benchmarks. The authors concluded that Logistic Regression provides the best balance of accuracy, performance, and explainability making it the best fit for practical implementations.

3) *Deep Learning Models Performance and Optimization:* Deep learning (DL) models offer superior capabilities in extracting complex patterns from data but are inherently more resource-intensive than traditional machine learning methods. Their success in cybersecurity applications, including phishing detection, has led to the development of numerous DL-based solutions [20]. Models such as BERT, DistilBERT, and ANN variants have demonstrated high accuracy levels across datasets like Enron and Lingspam, often exceeding 98%. However, these gains come at the cost of significantly larger model sizes, with baseline BERT models containing over 100 million parameters. Given the emphasis on cost-efficiency in this study, we deliberately avoided using large pre-trained models. Our approach instead utilized lightweight architectures totaling 4.7 million parameters, offering a substantial reduction in computational demand while maintaining strong predictive performance. This design choice was inspired by the work [20], which demonstrated that augmenting a 1D-CNNPD model with LSTM, GRU, and their bi-directional variants could achieve high accuracy without excessive resource requirements. Our methodology builds on these findings to strike a balance between performance, scalability, and deployment feasibility.

4) *Small Vs Large LLMs:* The results below, comparing the small LLMs to Claude, show that there is significant room for improvement for the small LLMs. While not a 1-1 comparison due to different testing parameters, it still shows that with an accuracy of up to 80%, small LLMs are capable

of understanding if emails have phishing content. An LLM trained specifically on phishing datasets would understand phishing emails much better, as its parameters would be trained on phishing data rather than irrelevant information, such as the ability to code.

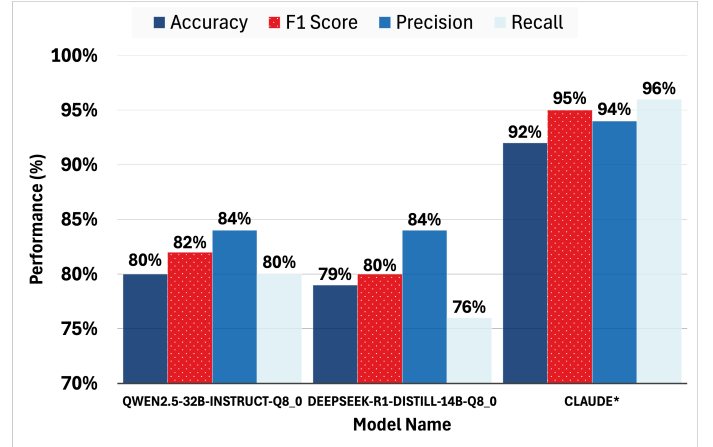


Fig. 4. Performance comparison between Full & Small LLMs. The bar marked with \* indicates the result for Claude taken from [12].

As shown in Fig. 4, quantized small LLMs can get close to a large model’s phishing-detection performance. For example, Qwen 2.5 32B Q8 achieves 81% accuracy, which is only 12% below the significantly larger Claude model. Although these results are obtained from slightly different test sets, they still demonstrate that a modern, carefully tuned LLM would be capable of delivering a highly accurate phishing detection model while requiring fewer resources.

### C. Current Research Gaps

Despite notable progress in ML, DL, and LLM-driven phishing detection, several critical gaps persist that limit the robustness, adaptability, and real-time deployment of these approaches.

1) *Lack of Specialized Phishing Detection Models:* Current LLMs are general-purpose and not optimized for phishing-specific patterns. Training LLMs exclusively on phishing-related data could enhance sensitivity to subtle deception tactics such as spear phishing and lateral attacks [8].

2) *Deployment Inefficiencies and Quantization Needs:* Large LLM-based phishing models are resource-intensive, which hinders their deployment in real-time and at the edge. Quantizing models into smaller, energy-efficient variants is crucial to balance performance and operational cost [8].

3) *Outdated and Limited Datasets:* Existing phishing datasets are often static, brand-specific, and fail to capture evolving attack vectors. Updated datasets, incorporating LLM-generated phishing samples and adversarial variants, are essential for developing resilient models [9].

4) *Incomplete Hybridization of Detection Techniques:* Standalone LLMs could be more effective when integrated with traditional rule-based or ML classifiers. Hybrid architectures can leverage the adaptability of LLMs and the precision

of classical methods to improve resilience against evolving threats [9].

5) *Limited Research on Adversarial Robustness:* The adversarial vulnerabilities of phishing detection LLMs remain underexplored. Investigations into adversarial training, red-teaming, and watermarking techniques are crucial for enhancing their reliability in attack scenarios [8].

6) *Insufficient Early Detection of Malicious Prompts:* Recent studies show promise in detecting malicious prompts that manipulate LLMs into generating phishing content [12]. Early intervention mechanisms can prevent automated phishing attacks and reduce the need for complex downstream detection, substantially improving system efficiency.

### III. PROPOSED MODEL

#### A. Methodologies

1) *Small LLMs:* The evaluation system for testing small LLMs is implemented using a Python script that loads a subset of phishing emails from the dataset into a CSV file. It automatically discovers the GGUF-formatted LLM checkpoint with the MODEL\_FILES mapping. Using llama-cpp-python, each model is initialized with the appropriate number of GPU layers and an 8,000-context window. The script has two versions, for the thinking and non-thinking models. The thinking models can complete their thinking portion, which may take some time due to the number of tokens it requires. In contrast, the normal models are only given 100 tokens to respond with, which is enough for a True or False response and a brief explanation. For the tests conducted in this paper, the LLM's explanation is not required, and thus, the LLMs are cut off early to save time during testing, which is part of the reason the thinking models have longer run times in our tests. The script then calls the LLM and provides a prompt with instructions and the email's body.

#### Test Prompt Text

You are an AI email security assistant specializing in phishing detection. Your task is to analyze the email below and decide whether it is a phishing or scam attempt.

"Please follow these strict instructions:"

"1. Your response must be a single line starting with either 'TRUE:' or 'FALSE:'"

"2. 'TRUE:' means you believe the email is a phishing/scam attempt; 'FALSE:' means you do not."

"3. Immediately after the colon, provide a concise explanation for your decision (do not include any extra text or commentary)."

"For example, if the email asks you to click a suspicious link for account verification, you might respond:"

"TRUE: The email contains urgent language and a suspicious link, typical of phishing attempts."

"Now, analyze the following email:"

"Email Content: ..."

Metrics such as total generation time, time to first token, and tokens per second are recorded at the end of every response. Using a regular expression (regex) parser, the first valid classification of True or False is identified and recorded. If none are found, then the response is considered "Failed." Every result is saved periodically to a CSV file to track the raw data for each response. When testing is completed, the scikit-learn library calculates two metric sets: one that includes all failures and one that excludes them, including accuracy, precision, recall, F1 Score, and AUC-ROC. Once the program is completed, the remaining raw data is appended to the raw data CSV file, and the final results are saved as a normal text file. All the LLM tests were run using Python 3.13.2 with the following libraries llama\_cpp\_python 0.3.8, pandas 2.2.3, and scikit-learn 1.6.1.

2) *Machine Learning:* The model evaluation pipeline was implemented using Python with libraries such as Pandas, NumPy, scikit-learn, and Matplotlib. The dataset was imported from the phishing email dataset CSV file, labels were binarized, and the data was split into training and testing sets. Text inputs were vectorized using TF-IDF with a 1000-feature limit and English stop-word removal. Logistic Regression, Naive Bayes, Random Forest, and SVM were trained and tested on the transformed data. The models were evaluated using accuracy, precision, recall, F1-score, AUC, and prediction time per 1000 emails. ROC curves were plotted for each model. The more robust the model in accuracy and speed, the more suitable it was for real-time phishing detection scenarios.

3) *Deep Learning:* Almost all the libraries used in the Machine Learning experimentation were also used in the Deep Learning approach. The same dataset and a relatively similar preprocessing pipeline were used to ensure the ultimate outcomes were more directly comparable. The maximum token limit for vectorization was set to 20,000, and padding was introduced to create a uniform input size for the neural networks. A factory class was designed to generate the model layer definitions, and functions were developed to iteratively run the models, test them, collect model metrics, and create visualizations, such as confusion matrices and ROC curves. Finally, each model was run five times to calculate the average expected performance, thereby avoiding biased results from random spikes, as the model's intended use is for practical applications. The Keras library was used to create the RNN models, and the training was conducted on two separate platforms. The training time was relatively short, with a total training time of less than 10 minutes for the four models. Even the most resource-intensive model, the Bi-Directional LSTM model, took only approximately 200 seconds to complete training. A Second training set was conducted on Google Colab to allow comparisons with the ML models' training and inference time. Multiple rounds of optimizations were performed on the code to ensure minimal resource waste.

#### B. Dataset & Resources

The dataset used for testing is a classification dataset containing 83,446 emails, sourced from the 2007 TREC Public

Spam Corpus and the Enron-Spam Dataset. It includes labeled data, where 1 indicates spam and 0 indicates legitimate emails for any given body text. The combined dataset was curated and made publicly available by Purusinghvi on Kaggle [21]. All the models were trained on a MacOS Sequoia 15.3.2 system using an M4 Max with 64GB of RAM, 48GB usable as VRAM.

### C. Results & Analysis

1) *Small LLM Performance:* Across the tests, the quantized small LLMs showed a wide level of performance variation. The DeepSeek R1 Distill Qwen models outperformed their same-size non-thinking Qwen 2.5 counterparts, achieving up to 79% accuracy in the 14B variant, versus the 55% accuracy for the Qwen 2.5 14B. This result suggests that small thinking LLM models may offer substantially better performance for phishing detection compared to their normal counterparts. However, they also take longer to output an answer, as shown in the runtime variation between the two types of models. This underscores that higher parameter counts do not necessarily mean a substantially better result. When the invalid responses are also included as wrong, the drop in performance is substantial in all models, but the 14B DeepSeek R1 Distill Qwen model had a lower drop in accuracy at 4%, compared to Qwen 32B’s drop of 10%. These results show that even without fine-tuning existing models, small LLMs such as DeepSeek R1 Distill Qwen 7B Q4\_K\_M are capable of detecting phishing emails at a higher rate than guessing, suggesting there is considerable room for improvement in a specially trained phishing detection model. When invalid responses are included, the other core metrics (F1 score, AUC-ROC, and Recall) exhibit commensurate declines. The smallest models, Qwen 2.5 7B, and 14B variant, performed very poorly with accuracies as low as 49% and 55%, akin to flipping a coin. This poor performance is likely due to not enough phishing related data in their parameters. In comparison, the distilled DeepSeek R1 7B Q4\_K\_M model had a 72% Accuracy, a 76% F1 score, 72% Precision, and 82% Recall while only using around 5 GB of VRAM. When invalid responses were included, the model still had a higher level of accuracy at 63%, which is a 9% drop. The LLMs were also able to explain why an email may or may not be phishing, in addition to the TRUE/FALSE classification. For example: "FALSE: The email appears to be a legitimate weather advisory from the National Weather Service, providing information about coastal hazards and surf conditions without requesting personal information or including suspicious links." DeepSeek R1 Distill Qwen 14B, in this instance, not only correctly labeled the sample as non-phishing but also provided a good summary of why the email is likely legitimate. Providing such explanations to users by highlighting key indicators or a lack thereof can reduce the likelihood of a person being fooled by a phishing email. The resource trades offs are high, for instance Qwen 32B used around 127% more VRAM than DeepSeek R1 Qwen 14B, but offered only a small 2% increase in accuracy (81% vs 79%). While the deepseek generated tokens faster, it still ran

significantly slower since it needed more tokens overall to think with 5.95 hours vs 2.67 hours, a 122.8% increase in time. The “Thinking” models took longer to complete in our testing setup because we let them generate their full chain of thought before outputting a True/False decision. In contrast, the standard models were cut off a few tokens after their True/False response. This was done to limit runtime due to hardware and time constraints. Allowing the reasoning models to complete their internal thinking steps naturally added a significant amount of time to the testing, which explains part of the duration difference. DeepSeek R1 Distill Qwen is more capable in VRAM-constrained scenarios, as the 14B model only requires around 15 GB of VRAM, while the 32B version of Qwen uses more than double that, at around 34 GB. This shows several trade-offs between each model that need to be considered when implementing LLMs in a phishing detection system. Overall, both the 32B Qwen and 14B DeepSeek R1 Distill Qwen models demonstrate significant potential for phishing detection, remarkably when fine-tuned to enhance their performance. The Tables I and II illustrate the same set of results from the LLMs, however Table II includes invalid responses. When invalid outputs are excluded (Table I), each model’s Accuracy and F1 score are higher—for example, Qwen 2.5 32B Q8 achieves 81% accuracy (F1 0.82). Including those invalid responses as errors (Table II) uniformly lowers performance. Qwen 2.5 32B Q8 drops to 71% accuracy (F1 0.73), with similar 5 to 10 point declines seen in the other models. This demonstrates further prompt tuning, and LLM training would likely help improve the instruction-following capabilities of these models. One limitation is that the LLMs were not finetuned on the dataset, meaning the parameters aren’t optimized to find nuanced linguistic patterns. Fine tuning the models on a phishing specific dataset would likely provide substantial performance improvements and allow for a better comparison against the ML and DL models.

2) *Deep Learning Performance:* Each of the proposed models is developed by augmenting the 1D-CNNPD layer with LSTM, GRU, Bi-LSTM and Bi-GRU as suggested by the work [20]. Additionally, the models were tested with a leaky ReLu activation in the fully connected layer, as the initial outcomes in the training epochs displayed stagnation and subsequent degradation. Note that all the results shown and plotted were generated from an average of 5 subsequent model executions and evaluations. As apparent from the ROC curve in Fig. 5, even the regular Deep Learning models provide a significant performance boost over regular Machine Learning models, with almost all the models displaying an AUC over 0.998. There are some marginal gains among the models as well, with Bi-Directional LSTM performing the best in terms of accuracy (98.74%) and GRU performing the worst (98.62%), although only slightly. Although this is acceptable, the loss metrics during the training epochs showed room for improvement.

During training, it was observed that performance gains began to plateau after the seventh epoch. One of the possible diagnoses was dying neurons, which is a common problem when using ReLu activation. The dying ReLU refers to a

TABLE I  
COMPARATIVE ANALYSIS OF AI MODELS FOR PHISHING DETECTION (INVALID RESPONSES EXCLUDED)

Model	Accuracy	F1 Score	Precision	Recall	AUC-ROC	Avg. TPS	Runtime (hr)	VRAM Usage
Qwen 2.5 7B Q4	0.49	0.34	0.58	0.24	0.52	27.94	0.54	5 GB
Qwen 2.5 14B Q8	0.55	0.39	0.74	0.27	0.58	11.97	0.84	15 GB
Qwen 2.5 32B Q8	0.81	0.82	0.84	0.80	0.81	6.54	2.67	34 GB
DeepSeek R1 Distill Qwen 7B Q4	0.72	0.76	0.72	0.82	0.71	35.00	4.08	5 GB
DeepSeek R1 Distill Qwen 14B Q8	0.79	0.80	0.84	0.76	0.79	15.16	5.95	15 GB

TABLE II  
COMPARATIVE ANALYSIS OF AI MODELS FOR PHISHING DETECTION (INVALID RESPONSES INCLUDED)

Model	Accuracy	F1 Score	Precision	Recall	AUC-ROC	Avg. TPS	Runtime (hr)	VRAM Usage
Qwen 2.5 7B Q4	0.45	0.31	0.51	0.22	0.48	27.00	0.54	5 GB
Qwen 2.5 14B Q8	0.52	0.37	0.68	0.25	0.55	11.71	0.84	15 GB
Qwen 2.5 32B Q8	0.71	0.73	0.76	0.69	0.71	6.34	2.67	34 GB
DeepSeek R1 Distill Qwen 7B Q4	0.63	0.68	0.65	0.71	0.62	33.83	4.08	5 GB
DeepSeek R1 Distill Qwen 14B Q8	0.75	0.76	0.81	0.72	0.76	14.94	5.95	15 GB

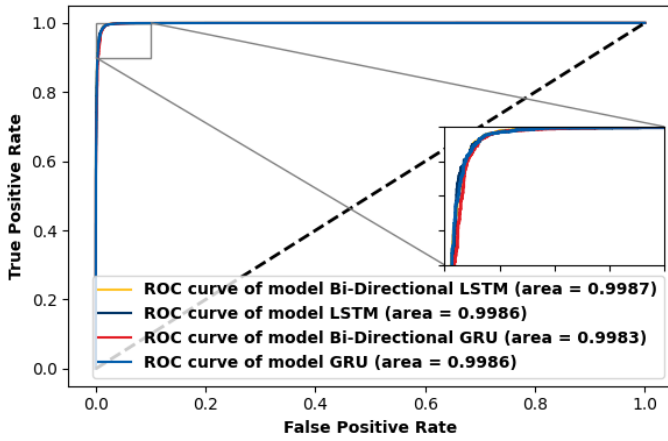


Fig. 5. ROC curves for Deep Learning models

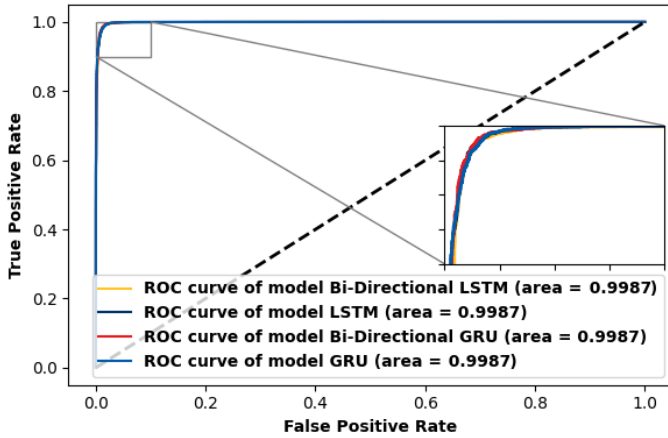


Fig. 6. ROC curves for Deep Learning models with Leaky ReLU Activation situation in which ReLU neurons become inactive due to their lower limit of 0. There are many empirical and heuristic explanations of why ReLU neurons die. However, little is known about its theoretical analysis [22]. To remedy this issue, we used the suggested Leaky ReLU activation function. As

shown in Fig. 6, this improvement enhanced the overall model performance, particularly for GRU-based models. Table III supports the outcome displayed in Fig. 6, Leaky ReLU activation provides marginal performance gains for most models. It produces the best performing model so far, the Bi-Directional GRU, with an accuracy of 98.77% and an AUC of 0.9987 (non-rounded). As a GRU model, it also has shorter training and inference times. This model was the most optimal in terms of cost-to-performance among all the models tested.

TABLE III  
MODEL PERFORMANCE COMPARISON: ReLU vs Leaky ReLU ACTIVATION

Model	ReLU Activation		Leaky ReLU Activation	
	Accuracy	ROC-AUC	Accuracy	ROC-AUC
LSTM	0.98664	0.99859	0.98706	0.99872
Bi-LSTM	0.98736	0.99875	0.98730	0.99866
GRU	0.98628	0.99824	0.98748	0.99871
Bi-GRU	0.98748	0.99857	0.98766	0.99874

3) *ML Performance:* Among the evaluated models for phishing detection, Random Forest stands out as having the highest overall performance in our tests. Fig. 7 and Fig. 8 show that it achieves top scores across all metrics, accuracy (0.9801), precision (0.9862), recall (0.9718), and F1-score (0.9789), indicating strong reliability and minimal error rates. However, its prediction time of 0.0276 seconds per 1000 emails, though reasonable, is higher compared to faster models. Logistic Regression offers a more balanced solution, combining solid performance accuracy of 0.9704, F1-score of 0.9686, and an unmatched prediction speed of 0.0001 seconds. This makes it especially suitable for real-time or large-scale deployments where speed and accuracy are essential. It remains a practical and efficient choice for phishing detection systems. Naive Bayes, with the lowest accuracy (0.9466) and F1-score (0.9435), excels in speed, processing 1000 emails in only 0.0002 seconds. It is ideal for constrained environments requiring high throughput, but can be less reliable in terms of Precision. SVM performs well in terms of accuracy (0.9711)



and F1-score (0.9694); however, its major limitation is slow processing time, which takes 1.3934 seconds for 1000 emails, making it less practical for real-time applications.

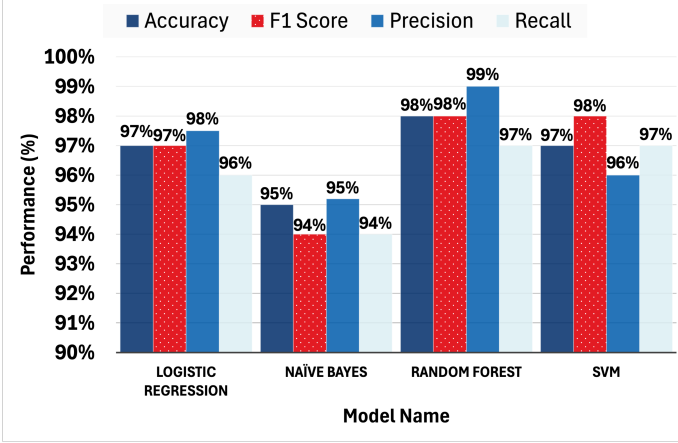


Fig. 7. Comparative results of Machine Learning Algorithms

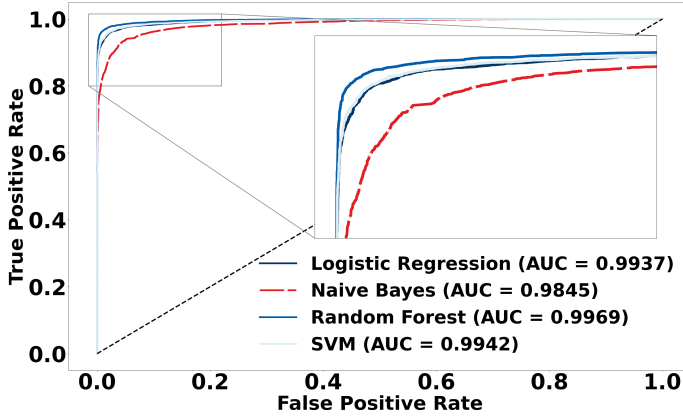


Fig. 8. ROC curves for all the tested ML models

In summary, Random Forest is the most accurate, Logistic Regression is the most balanced and scalable, Naive Bayes is the fastest, and SVM offers high precision but at a cost of efficiency.

4) *Resource Consumption*: One of the primary objectives of this research was to identify the overall resource consumption of each approach and the resultant performance gain. This idea was prompted by the massive reported costs of training and deploying LLMs, as well as the impact of such expenses on their applicability in practical scenarios.

Table IV shows the inference time for each of the LLM, ML, and DL models tested in the paper. The ML models are approximately 1308 times faster than the DL models in inference, with nearly the same performance, but with an average of 2% less accuracy. The ML models are also approximately 1.38 million times faster than the LLMs, illustrating that using LLMs to check every email is significantly less efficient. Instead, LLMs are best suited for complex cases that require contextual reasoning and human-readable explanations.

TABLE IV  
MODEL EFFICIENCY COMPARISON

Model Name <sup>a</sup>	Inference Time (s) <sup>b</sup>	Best Accuracy
Qwen 2.5 7B Q4	1944	49%
Qwen 2.5 14B Q8	3024	55%
Qwen 2.5 32B Q8	9612	81%
DeepSeek R1 7B Q4	14688	72%
DeepSeek R1 14B Q8	21420	79%
Logistic Regression	0.0001	97%
Naive Bayes	0.0003	95%
Random Forest	0.0033	98%
Linear SVM	0.0006	97%
GRU	0.3850	98.75%
Bi-GRU	1.4250	98.77%
LSTM	0.4400	98.71%
Bi-LSTM	1.4458	98.74%

However, in order to accommodate this sophisticated functionality, LLMs consume a massive amount of precious resources such as water and energy. Moreover, the resulting carbon footprint poses a significant concern. In a comprehensive research conducted by Jegham et al. [23], where 30 major LLMs were analyzed, it was found that the most resource-intensive LLMs, such as ChatGPT-O3 and DeepSeek R1, can consume up to 30Wh per long prompt. More significantly, the research estimates that with the measured 0.43Wh per query and the reported 700 million queries per day, annually, ChatGPT-4o uses energy comparable to approximately 35,000 US households and evaporates freshwater matching the annual drinking needs of 1.2 million people, concluding a chicao-sized forest would be needed to offset the carbon footprint. Wong V. [24] calculated that serving a single prompt in ChatGPT produces more than 4 grams of  $CO_2eq$ , which is over 20 times the operational carbon footprint of a web search query. The environmental impact becomes even more concerning when considering that findings published by Chien et al. [25] show the energy consumption of LLM serving has now surpassed that of training. This shift toward serving-dominated energy consumption creates a complex sustainability problem that requires comprehensive solutions such as hardware lifecycle management and energy efficiency optimization as proposed by Ding et al. [26]. The results from our tests align well with these observations: during inference, the 14 billion-parameter DeepSeek R1 Distill Qwen Q8\_0 model required approximately 6 hours of compute and 15 GB of VRAM to identify phishing emails. In comparison, the best-performing deep learning model Leaky ReLU Bi-GRU achieved better performance, with an accuracy of 98.77% versus 79% for DeepSeek R1 Distill Qwen 14B Q8\_0 in just 15.33 seconds, substantially less than the 5.95 hours DeepSeek R1 Distill Qwen 14B Q8\_0 took. While the testing platforms were not identical, the substantial difference in performance far exceeds what could be explained by hardware alone. Classical machine learning approaches performed similarly to DL, with models such as random forest requiring only a few seconds to run and providing around 98% accuracy. These findings suggest a hybrid approach for real-world deployments where ML and DL models handle the bulk of cases. At the same time, a fine-tuned small LLM intervenes in the

most challenging scenarios, optimizing accuracy, cost, and environmental impact simultaneously.

#### IV. CONCLUSION

This study evaluated phishing detection approaches across traditional ML, DL, and small LLMs. Our findings reveal that models like DeepSeek R1 Distill Qwen 14B achieve over 79% accuracy using only 15 GB of VRAM, while Bi-GRU models exceed 98% accuracy with minimal inference time, highlighting their cost-effectiveness. Despite their limitations in raw accuracy, LLMs provide contextual explanations and resilience against adversarial rephrasing, key advantages for real-time phishing detection. A hybrid framework that combines the strengths of ML/DL with context-aware LLMs can deliver improved accuracy, interpretability, and efficiency. Future work should focus on specialized LLM training using phishing-centric datasets, robust adversarial defence strategies, and real-time threat integration. This hybrid approach offers a practical path toward scalable, interpretable, and environmentally conscious phishing detection systems in the Gen-AI era. To improve future phishing detection, work should focus on developing specially trained LLMs that primarily use phishing-related content to enhance their detection of nuanced and context-driven attacks. Creating large, fully trained models for phishing detection and then quantizing them can also be done to strengthen computational requirements. The results from this paper are limited by the reliance on English-only sources in the dataset. This fails to account for linguistic and cultural diversity of global phishing attacks, incorporating modern multilingual datasets could produce a universally effective model. Finally, combining trained LLMs and traditional detection methods, such as ML, in a hybrid approach would be the most optimal overall scenario. Ultimately, focus should be on enhancing adversarial resilience and facilitating early detection of malicious prompts, thus significantly mitigating misuse while reducing computational costs.

#### REFERENCES

- [1] F. P. E. Putra, A. Zulfikri, G. Arifin, and R. M. Ilhamsyah, "Analysis of phishing attack trends, impacts and prevention methods: Literature study," *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 1, pp. 413–421, 2024.
- [2] Y. Otoum, P. Singh, and A. Nayak, "Advancing iomt defenses: Deep collaborative learning for robust healthcare security," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 2966–2971.
- [3] M. Alghenaim, G. Alkaws, and C. R. Barnhart, "The state of the art in ai-based phishing detection: A systematic literature review," *Studies in Computational Intelligence*, vol. 1178, 2025, in: Al-Sharafi, M.A., Al-Emran, M., Mahmoud, M.A., Arpaci, I. (eds), *Current and Future Trends on AI Applications*. Springer, Cham.
- [4] Y. Otoum, A. Asad, and A. Nayak, "Llm-based threat detection and prevention framework for iot ecosystems," *arXiv preprint arXiv:2505.00240*, 2025.
- [5] —, "Llms meet federated learning for scalable and secure iot management," *arXiv preprint arXiv:2504.16032*, 2025.
- [6] K. Afane, W. Wei, Y. Mao, J. Farooq, and J. Chen, "Next-generation phishing: How llm agents empower cyber attackers," *IEEE International Conference on Big Data (BigData)*, pp. 2558–2567, 2024.

- [7] J. Dodge, T. Prewitt, R. Tachet des Combes, E. Odmark, R. Schwartz, E. Strubell, A. S. Luccioni, N. A. Smith, N. DeCario, and W. Buchanan, "Measuring the carbon intensity of ai in cloud instances," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2022, p. 1877–1894.
- [8] M. Bethany, A. Galiopoulos, E. Bethany, M. B. Karkevandi, N. Vishwamitra, and P. Najafirad, "Large language model lateral spear phishing: A comparative study in large-scale organizational settings," *arXiv preprint arXiv:2401.09727*, January 2024. [Online]. Available: <https://arxiv.org/abs/2401.09727>
- [9] A. Kulkarni, V. Balachandran, D. M. Divakaran, and T. Das, "From ml to llm: Evaluating the robustness of phishing webpage detection models against adversarial attacks," *arXiv preprint arXiv:2407.20361*, July 2024. [Online]. Available: <https://arxiv.org/abs/2407.20361>
- [10] H. İş, "Llm-driven sat impact on phishing defense: A cross-sectional analysis," in *Proceedings of the 2024 12th International Symposium on Digital Forensics and Security (ISDFS)*, April 2024.
- [11] T. Quinn and O. Thompson, "Applying large language model (llm) for developing cybersecurity policies to counteract spear phishing attacks on senior corporate managers," May 2024.
- [12] S. S. Roy, P. Thota, K. V. Naragam, and S. Nilizadeh, "From chatbots to phishbots?: Phishing scam generation in commercial large language models," *IEEE Symposium on Security and Privacy (SP)*, pp. 36–54, 2024, presented at the IEEE Symposium on Security and Privacy, San Francisco, CA, USA.
- [13] P. H. Kyaw, J. Gutierrez, and A. Ghobakhloo, "A systematic review of deep learning techniques for phishing email detection," *Electronics*, vol. 13, no. 3823, Sep 2024.
- [14] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 4545, Nov 2023.
- [15] Y. Otoum, A. Asad, and I. Ahmad, "Open-source llm-driven federated transformer for predictive iot management," *arXiv preprint arXiv:2505.00651*, 2025.
- [16] K. Debnath and N. Kar, "Email spam detection using deep learning approach," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, Faridabad, India, 2022, pp. 37–41.
- [17] A. A. Ali and A. A. Abdullah, "Text email spam adversarial attack detection and prevention based on deep learning," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 2, 2025.
- [18] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv preprint arXiv:2009.11116*, Sep 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2009.11116>
- [19] F. Greco, G. Desolda, and L. Viganò, "Supporting the design of phishing education, training and awareness interventions: An llm-based approach," *CEUR Workshop Proceedings*, vol. 3700, pp. 8:1–8:12, 2024. [Online]. Available: <https://ceur-ws.org/Vol-3700/paper8.pdf>
- [20] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing phishing email detection: A comparative study of deep learning models," *Sensors*, vol. 24, no. 7, p. 2077, Mar 2024.
- [21] P. Purusinghvi, "Email spam classification dataset," <https://www.kaggle.com/datasets/purusinghvi/email-spam-classification-dataset>, 2023, accessed: April 2025.
- [22] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying relu and initialization: Theory and numerical examples," *Communications in Computational Physics*, vol. 28, no. 5, p. 1671–1706, Jan. 2020. [Online]. Available: <http://dx.doi.org/10.4208/cicp.OA-2020-0165>
- [23] N. Jegham, M. Abdelatti, L. Elmoubarki, and A. Hendawi, "How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference," 2025. [Online]. Available: <https://arxiv.org/abs/2505.09598>
- [24] V. Wong, "A closer look at the carbon footprint of chatgpt - piktochart," May 2024. [Online]. Available: <https://piktochart.com/blog/carbon-footprint-of-chatgpt/>
- [25] A. A. Chien, L. Lin, H. Nguyen, V. Rao, T. Sharma, and R. Wijayaradana, "Reducing the carbon impact of generative ai inference (today and in 2035)," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. New York, NY, USA: Association for Computing Machinery, 2023.
- [26] Y. Ding and T. Shi, "Sustainable llm serving: Environmental implications, challenges, and opportunities : Invited paper," in *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, 2024, pp. 37–38.