Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.075161

ARTICLE

A Comparative Analysis of Machine Learning Algorithms for Spam and Phishing URL Classification

Tran Minh Bao[1], **Kumar Shashvat**[2], **Nguyen Gia Nhu**[3,*] **and Dac-Nhuong Le**[4]

[1]HCMC University of Industry and Trade, Ho Chi Minh, 10000, Vietnam
[2]Amity School of Engineering & Technology, AMITY University, Bengaluru, 226028, India
[3]School of Computer Science, Duy Tan University, Danang, 55000, Vietnam
[4]Haiphong University, Haiphong, 05000, Vietnam

*Corresponding Author: Nguyen Gia Nhu. Email: nguyengianhu@duytan.edu.vn

Received: 26 October 2025; Accepted: 16 December 2025; Published: 12 March 2026

ABSTRACT: The sudden growth of harmful web pages, including spam and phishing URLs, poses a greater threat to global cybersecurity than ever before. These URLs are commonly utilised to trick people into divulging confidential details or to stealthily deploy malware. To address this issue, we aimed to assess the efficiency of popular machine learning and neural network models in identifying such harmful links. To serve our research needs, we employed two different datasets: the PhiUSIIL dataset, which is specifically designed to address phishing URL detection, and another dataset developed to uncover spam links by examining the wording and structure of every URL. Our strategy was to train and evaluate four classification models, namely Random Forest, Support Vector Machine (SVM), Naive Bayes, and Artificial Neural Networks (ANN), under two different feature engineering approaches: statistical text-based analysis and heuristic-based structural features. The results are in, and they are stunning: Random Forest and ANN models were always the best. During our research, we achieved some outstanding results. On the PhiUSIIL phishing dataset, the model achieved an accuracy of 99.99%, and on the spam dataset, it attained an accuracy of 99.62%. Studies surpass any previously reported findings, firmly establishing the efficacy of machine learning and neural networks in detecting malicious URLs. Not only does this work reinforce the superiority of these in-demand models, but it also sets a high bar for subsequent research and development in the field. In general, this provides the direction for building smarter, faster, and more precise tools that can spot online threats as they develop.

KEYWORDS: Web security; phishing; malicious URL; DOM analysis; transformer; GNN; evaluation; adversarial ML; LLM safety

1 Introduction

In today's digital landscape, the rapid growth of unsafe and deceptive web links has increased the vulnerability of routine online activities. Malicious URLs remain one of the most common tools used by cyber attackers to launch phishing, spam, and malware campaigns. These attacks pose risks not only to individuals but also to enterprises, often leading to financial losses, credential theft, and large-scale data breaches. Real-world cases illustrate the severity of the threat. For example, in a recent high-profile phishing incident, attackers impersonated a cloud-storage provider. They tricked thousands of users, many of whom were MIM (Mobile Internet Messaging) users, into entering their login credentials on a spoofed, mobile-friendly page. Similar events continue to occur globally, showing how attackers design URLs that appear

legitimate on mobile screens, where the domain is often truncated and the malicious part of the URL is hidden.

Cybercriminals increasingly rely on advanced social engineering, HTTPS abuse, URL shortening services, and homograph attacks, where visually similar characters are substituted to deceive users. To understand and mitigate such threats, researchers have developed several domain-specific frameworks and datasets. One of the most comprehensive among them is PhiUSIIL [1], which provides a rich set of structural, lexical, and behavioural features extracted from both safe and phishing websites. Frameworks like PhiUSIIL play a critical role in evaluating detection algorithms under realistic and challenging conditions.

Fig. 1 illustrates a realistic phishing workflow commonly observed in modern attack campaigns. The process typically begins with a malicious SMS or messaging link that conceals the true target using redirection or URL shortening. Once the user clicks the link, they are taken to a deceptive domain crafted through homograph tricks or visual similarity to a legitimate service. The fake page harvests the user's authentication details, which are then exfiltrated to an attacker-controlled C2 server via an encrypted communication channel.

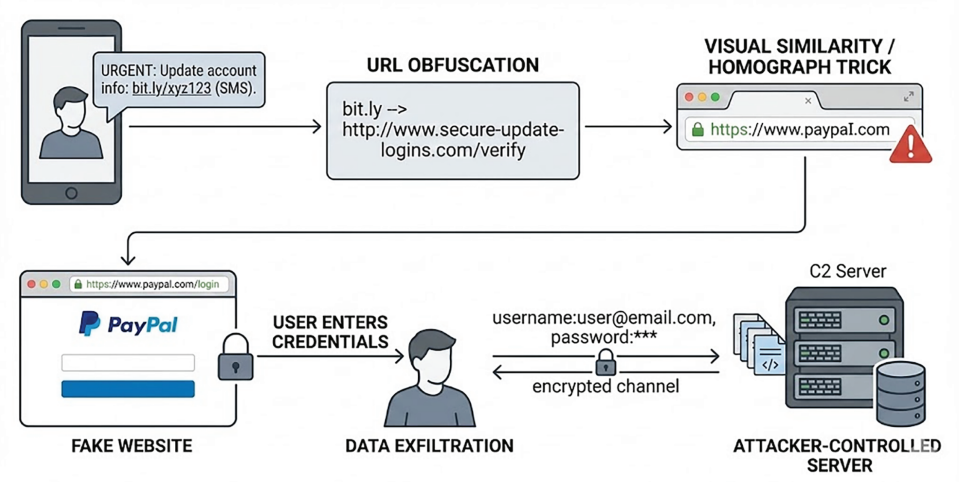

Figure 1: A real-world phishing URL attack chain

Machine learning (ML) and deep learning (DL) techniques have emerged as effective tools for automatically identifying malicious URLs. Prior studies have shown that models can differentiate between benign and malicious links using features related to URL composition, hosting behaviour, content properties, and character-level patterns. Although modern detection systems often achieve an accuracy of more than 95%, even small errors can have a significant real-world impact. For example, a security engine that scans one million URLs per day would still miss a thousand malicious links with just a 0.1% error rate. This highlights why high precision and robustness are essential in practical deployments.

Despite notable progress, several challenges remain. State-of-the-art transformer and graph-based models have improved detection performance, but they require substantial computational resources and complex training pipelines. They are often difficult to deploy in real-time, low-latency systems such as email gateways, mobile applications, and lightweight enterprise security appliances. Classical ML models like Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB), and Artificial Neural Networks (ANN) continue to be deployed in such environments due to their speed, interpretability, and ability to operate effectively with simpler feature representations.

Although malicious URL detection is a mature research area, there remains a need for unified, reproducible baseline studies that evaluate classical models under consistent conditions and across both lexical and structural feature engineering strategies. Modern work often examines only phishing or only spam URLs, leaving a gap in cross-domain benchmarking. Moreover, few studies provide head-to-head comparisons of classical models using distinct feature paradigms on widely used datasets such as the Spam URLs Classification Dataset and the PhiUSIIL phishing dataset.

Motivated by these gaps, this study provides a systematic comparison of four widely used machine-learning approaches, such as RF, SVM, NB, and ANN, applied to two complementary URL classification tasks: spam detection and phishing detection. The models are evaluated using both TF-IDF-based lexical features and heuristic structural features to understand how different representations influence performance. Our experiments achieve high accuracy rates on both tasks, demonstrating the continued relevance and effectiveness of classical models when applied in conjunction with appropriate feature engineering techniques.

The main contributions of this study are as follows:

1. A unified comparative analysis of four widely used machine-learning models, like Random Forest, SVM, Naïve Bayes, and ANN, evaluated consistently across two URL-classification tasks: spam detection and phishing detection.
2. A dual feature-engineering framework that examines how statistical lexical features (TF-IDF) and heuristic structural features influence model performance, providing insights into the strengths and limitations of classical ML approaches.
3. Benchmarking on two complementary datasets, Spam URLs Classification Dataset and PhiUSIIL, along with additional validation on recent phishing samples to demonstrate the robustness and generalisation capability of the models.
4. Comprehensive evaluation using multiple metrics, including Accuracy, F1-Score, ROC-AUC, and computational efficiency, enabling a transparent comparison with existing state-of-the-art studies.
5. Establishment of a strong and reproducible baseline, offering a practical reference point for future research incorporating transformer-based, graph-based, or multimodal URL-detection architectures.

2 Literature Review

The rapid increase in harmful web content, especially phishing and spam links, has become a major and ongoing concern for cybersecurity. These types of dangerous URLs are often used to trick people, spread viruses or malware, and steal sensitive information. Because of this, being able to spot and stop them quickly and accurately is extremely important. Over time, this has turned into a kind of back-and-forth battle, while security experts work hard to create smart detection systems, cyber attackers constantly come up with new ways to hide or disguise their harmful links.

To tackle this growing problem, many researchers have started using machine learning (ML) and deep learning (DL) technologies to automatically identify and classify suspicious URLs. This approach helps make the detection process faster and more efficient. In this review, we take a closer look at the latest progress in this area. We explore the different types of datasets that have been used, the various techniques and tools applied, and how well past studies have performed. This background helps explain and support the purpose and value of our own research.

2.1 Machine Learning Approaches for Phishing Detection

Researchers have spent a lot of time studying how to detect phishing URLs using different machine learning models. These models usually depend on carefully picking out features from the URLs, which is

known as feature engineering. The features are often taken from the way the URL is written (called lexical features), information about the host or server it comes from, and even the content of the webpage itself.

Earlier studies showed that this method works quite well, although it doesn't always give perfect results. For example, in 2018, Mahajan and Siddavatam [2] used a manual process to extract features from URLs and were able to reach an accuracy of 97.14%. In a more recent study, Jalil et al. [3] used a technique called TF-IDF to analyze parts of the URL and applied a Random Forest model, achieving an accuracy of 96.85%. Likewise, Alsariera et al. [4] used a specially improved algorithm called ForestPA (Forest by Penalizing Attributes), which helped them get to a 97.4% accuracy.

These results show that carefully designed features and smart models can be very effective in identifying phishing links, even though there's still some room for improvement.

The release of the PhiUSIIL dataset has been a big step forward in the study of phishing URL detection. This dataset includes a large and detailed collection of both safe (benign) and harmful (phishing) URLs, along with a wide range of features. Because of its size and complexity, it provides a more difficult and standardized way to test and compare different detection models. Many researchers have used this dataset and achieved impressive results, pushing the limits of how accurately phishing URLs can be detected.

For example, Etem and Teke introduced a new method that used a technique called t-SNE to reduce the number of features without losing important information. Using this approach with a Decision Tree model, they were able to reach an accuracy of 99.7%. Even though this is a very high score, it still shows that no model is perfect and there's always room for improvement [5].

In another study, Vajrobol et al. [6] focused on selecting only the most important features. They used a method based on mutual information to pick the top five most useful features and applied a Logistic Regression model. Surprisingly, even with just these few features, they achieved an outstanding accuracy of 99.97% While this result is incredibly close to perfect, there's still a tiny chance of errors, something our research aims to improve even further.

In addition, many researchers have turned to ensemble methods, which combine multiple models to make stronger predictions. One such framework, developed by Gałka et al., managed to detect phishing URLs with a true positive rate of 99.52%, while also keeping false alarms low. This shows that ensemble models are very powerful, but even they aren't flawless [7].

2.2 Deep Learning Innovations in URL Classification

Although traditional machine learning (ML) models have shown good results in detecting phishing URLs, deep learning models bring an extra advantage, they can automatically learn complex patterns from the data without needing a lot of manual effort to create features. This ability to "learn on their own" has sparked a new wave of interest in using deep learning for URL classification.

For instance, Ren et al. [8] used an advanced model called a Bi-directional LSTM (BiLSTM) combined with an attention mechanism and Word2Vec embeddings. This approach helped their model understand the structure and meaning of URLs better, leading to an accuracy of 98.06%. Similarly, Lakshmi and colleagues experimented with a Deep Neural Network (DNN) and used the Adam optimizer, reaching an accuracy of 96.00% [9]. Transformer-based architectures have recently gained traction. Patra et al. [1] evaluated several pre-trained BERT variants and found that the base model performed best, achieving an accuracy of 99.29%. FCNN models have also demonstrated strong performance, as shown in Rawla et al. [10], who achieved a 99.38% accuracy on the PhiUSIIL dataset. Although deep learning approaches often achieve high accuracy, surpassing the 99.9% threshold remains challenging, especially on datasets like PhiUSIIL.

2.3 Spam and General Malicious URL Detection

The task of classifying spam URLs is quite similar to phishing URL detection, as both often use the same kinds of features, mainly based on the structure of the URL (lexical features) and details about the host. For example, Jilani and Sultana worked with a spam URL dataset from Kaggle and found that a Random Forest classifier performed better than other models like Support Vector Machine (SVM) and Naïve Bayes. Their model reached an accuracy of 97.39%, once again showing that ensemble models, which combine the power of multiple learners, are very effective in this field [11].

In another study, Ubing et al. also relied on ensemble techniques and tested their approach on the UCI dataset. They were able to achieve a maximum accuracy of 95.4%, which further supports the value of using ensemble models for spam detection [12].

Deep learning has also shown strong results in detecting spam URLs. Kotni et al. experimented with several popular transformer-based models, including BERT, DistilBERT, and ALBERT. Among them, DistilBERT stood out as the best performer, reaching an accuracy of 95.02% in classifying spam URLs [13].

When it comes to broader malicious URL detection where not just spam and phishing, but also malware and website defacement are considered, the classification task becomes even more complex. This is because these different types of threats can have different patterns and behaviors. Jaswanthi et al. took on this challenge by running a comparative study using lexical analysis and several machine learning models on a diverse dataset. Their results showed that the Extra Trees classifier performed the best, achieving an accuracy of 92.01% [14].

In another notable work, Sahingoz et al. developed a system based on deep learning that used natural language processing (NLP) features to detect malicious URLs and achieved strong performance [15]. Similarly, Zamir et al. explored a variety of machine learning models to detect phishing websites, and their most successful model reached an accuracy of 97.00% [16].

Altogether, these studies make it clear that while very high accuracy is possible when focusing on specific types of malicious URLs, it becomes much harder to maintain such high performance when dealing with multiple types of threats at once. This highlights the ongoing challenges in building models that can generalize well across a wide range of cyber threats.

2.4 Summary and Positioning

The literature shows that both advanced machine learning models and deep learning architectures can achieve high accuracy in URL classification, with many results exceeding 95%. Yet, surpassing 99.9% remains challenging, especially on realistic datasets like PhiUSIIL. Prior work has established strong performance, but substantial gaps persist, particularly in achieving consistent performance across both phishing and spam tasks [17–19].

Our study builds on these foundations by evaluating Random Forest, SVM [20], Naïve Bayes [21], and ANN models [22] on two complementary datasets: one focused on spam URLs and the other on phishing URLs. Our results—99.62% accuracy on the spam dataset and 99.99% on the PhiUSIIL dataset—demonstrate that classical models can still deliver competitive performance. These findings contribute meaningful evidence supporting the use of machine learning and neural networks in practical cybersecurity applications.

Recent research in phishing and spam URL detection has also shifted toward deep, transformer-based, and graph-driven architectures capable of learning complex structural and contextual patterns in URLs. Post-2023 studies introduced models such as DomURLs-BERT, URLBERT [23], and PMANet, with PMANet using a post-trained language-model–guided attention mechanism to capture subtle malicious signals. Aljofey et al. proposed a multi-channel TCN-fusion approach combining URL and HTML content,

reflecting the growing interest in multimodal analysis. Graph Neural Networks (GNNs) have been applied to domain and URL graphs, enabling the detection of coordinated attack networks [24]. The survey by Tian et al. [25] synthesises these advances and highlights the importance of adversarial robustness and zero-day URL detection.

Even with these developments, many state-of-the-art systems still require substantial computational resources, extensive pretraining, or complex feature extraction workflows, making them challenging to deploy in real-time environments. Classical models, such as Random Forest, SVM, Naïve Bayes, and ANN, continue to be widely used due to their speed, interpretability, and efficiency, underscoring the ongoing need to evaluate their performance under consistent feature engineering strategies.

Table 1 summarises representative machine-learning, deep-learning, transformer-based, and graph-based approaches for phishing, spam, and malicious URL detection. It highlights feature types, datasets used, accuracy levels, and the key insight of each study, providing a structured overview of advancements in URL-based threat detection.

Table 1: Summary of major studies in phishing, spam, and malicious URL detection

Study	Model/ Approach	Feature type	Dataset	Accuracy (%)	Key contribution
Mahajan and Siddavatam (2018) [2]	Classical ML	Lexical + Handcrafted	Private	97.14	Manual features are effective for early phishing detection
Jalil et al. (2021) [3]	Random Forest + TF-IDF	Lexical	Custom	96.85	TF-IDF is competitive for URL text patterns
Alsariera et al. (2021) [4]	ForestPA (Ensemble)	Lexical/Heuristic	Private	97.4	Penalisation-based ensemble improves phishing accuracy
Etem and Teke (2021) [5]	Decision Tree + t-SNE	Lexical	PhiUSIIL	99.7	Dimensionality reduction aids classical ML
Vajrobola et al. (2024) [6]	Logistic Regression	Top-5 MI Features	Private	99.97	Minimal features can achieve near-perfect accuracy
Gałka et al. (2024) [7]	Ensemble ML	Lexical/ Heuristic	Multiple	99.52	Strong and stable with low false positives
Ren et al. (2019) [8]	BiLSTM + Attention	Sequence	Private	98.06	Deep models learn URL structure automatically
Lakshmi et al. (2021) [9]	DNN	Lexical	Private	96.0	DNN is competitive with ML on small datasets
Patra et al. (2024) [1]	BERT Variants	Token/Character Embedding	Public	99.29	Transformers outperform classical ML

(Continued)

Table 1 (continued)

Study	Model/ Approach	Feature type	Dataset	Accuracy (%)	Key contribution
Rawla et al. (2025) [10]	FCNN	Lexical/ Structural	PhiUSIIL	99.38	DL excels on complex URL attributes
Jilani and Sultana (2022) [11]	Random Forest	Lexical/Host	Kaggle Spam	97.39	RF is strong for spam classification
Ubing et al. (2019) [12]	Ensemble ML	Lexical/Host	UCI	95.4	Ensemble ML is robust across datasets
Kotni et al. (2022) [13]	DistilBERT	Transformer	Spam Dataset	95.02	Lightweight transformer effective
Jaswanthi et al. (2024) [14]	Extra Trees	Lexical	Multi-threat	92.01	Multi-threat harder than single threats
Sahingoz et al. (2024) [15]	NLP-based DL	Sequence	Private	High	NLP captures semantic URL cues
Zamir et al. (2020) [16]	Classical ML	Lexical/ Structural	Phishing Websites	97.00	ML is still practical for multi-type threats
DomURLs-BERT (2023) [23]	Transformer	URL Embedding	Public	~99	Strong semantic modelling for URL strings
Ren et al. (2024) [8]	Transformer + Multi-Level Attention	Token/ Character	Public	~99	Captures subtle malicious cues
Prasad et al. (2024) [18]	Multimodal TCN + URL + HTML	Multimodal	Public	High	Multi-channel fusion resists modern attacks
Tian et al. (2025) [25]	GNN	URL graph topology	Public	High	Detects coordinated domain-based attacks

Identified Research Gap: Despite the progress described above, several issues remain unaddressed. Most studies focus only on phishing or spam, limiting cross-domain generalisation. Few works provide direct comparisons between classical ML models and modern deep architectures under identical experimental conditions, making it difficult to assess accuracy, latency, and computational trade-offs. Many approaches rely solely on either lexical or structural features, without examining how these representations interact across different model types. Furthermore, existing studies rarely address real-world complications such as evolving phishing techniques, shortened URLs, adversarial obfuscation, or truncated mobile URL displays.

To address these gaps, the present study offers a unified comparative analysis of Random Forest, SVM, Naïve Bayes, and ANN models using both lexical (TF-IDF) and structural heuristic features. Experiments are conducted on the Spam URLs Classification Dataset and PhiUSIIL datasets, with additional validation on recent phishing samples. By analysing accuracy, F1-Score, ROC-AUC, and computational efficiency, this work provides a transparent baseline for assessing the strengths and limitations of classical models relative to emerging transformer-based and graph-driven approaches.

3 Proposed Methodology

Our objective is to build a reliable and practical system for detecting malicious URLs. To evaluate the effectiveness of classical machine-learning models in different real-world scenarios, we use two benchmark datasets: the spam URLs classification dataset for spam URL detection and the PhiUSIIL dataset for phishing URL detection. Four widely used classifiers—Random Forest, Support Vector Machine (SVM), Naïve Bayes, and Artificial Neural Networks (ANN)—are assessed under two feature-engineering strategies: lexical analysis and structural/heuristic analysis. The overall workflow is shown in Fig. 2.

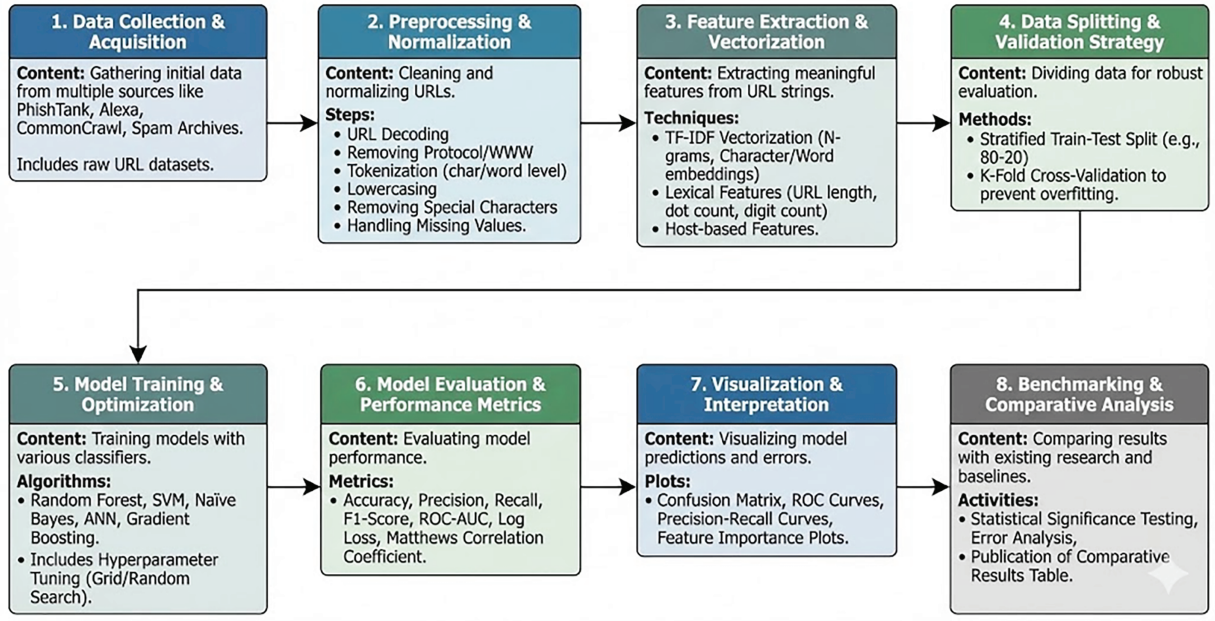

Figure 2: End-to-end workflow for URL classification

Fig. 2 provides an overview of the complete pipeline used in this study. The process begins with data collection, where URL samples are gathered from publicly available repositories containing spam and phishing examples. Once collected, all URLs undergo a uniform preprocessing step that includes URL decoding, removal of protocols and repeated prefixes (e.g., "http", "https", "www"), character- or word-level tokenisation, lowercasing, removal of special characters, and handling of missing entries. Shortened URLs (such as bit.ly, tinyurl, or goo.gl) are expanded by resolving their redirection chains. In cases where expansion is not possible due to expiration or intentional obfuscation, the shortened URL itself is used and processed using lexical and structural fallback features. Although URLs are often truncated on mobile screens for display purposes, the classification system processes the full underlying URL string obtained during data collection; therefore, user-interface truncation does not affect model performance.

After preprocessing, the feature-extraction stage is applied according to the type of dataset. For the Spam URLs Classification Dataset, we use lexical analysis and convert each URL into a TF-IDF representation derived from character- and word-level n-grams. This captures meaningful textual patterns such as length, token distribution, and character sequences that are common in spam links. For the PhiUSIIL dataset, we use the 56 structural and heuristic features provided in the benchmark, which include URL-based, domain-based, and content-related indicators. Continuous features are standardised to zero mean and unit variance using parameters computed only from the training split.

The processed data is then divided into training and testing sets using a stratified split to preserve class balance. K-fold cross-validation is employed to ensure robust evaluation and to prevent overfitting. The four selected machine-learning models—Random Forest, SVM, Naïve Bayes, and ANN—are trained on both feature-engineering strategies. Where appropriate, hyperparameters are tuned using grid search or random search optimisation to improve performance.

Model performance is assessed using standard evaluation metrics, including Accuracy, Precision, Recall, F1-Score, ROC-AUC, and Log Loss. To improve interpretability, we visualise predictions and errors through confusion matrices, ROC curves, precision–recall curves, and feature-importance plots. The final stage involves benchmarking our results against previously published work. This includes statistical significance testing, error rate comparison, and the preparation of comparative performance tables to contextualise our findings within the existing literature.

This structured methodology ensures a transparent, reproducible, and comprehensive evaluation of classical machine-learning models for both phishing and spam URL detection.

3.1 Datasets

3.1.1 Dataset 1: Spam URLs Classification Dataset

The first dataset is designed for a binary classification task, identifying whether a given URL is spam or not. It includes two main pieces of information for each entry: the raw URL string and a label indicating whether the URL is spam (originally provided as a boolean value, which we converted into integers, 1 for spam, and 0 for legitimate) [17].

This dataset is particularly well-suited for testing how well models can learn from the structure and text of URLs alone. Since it focuses on the lexical content of URLs, it helps us evaluate the ability of our models to recognize patterns and keywords that typically appear in spam links.

This dataset is ideal for evaluating how effectively models can identify spam based solely on a URL's structure and text. Because it focuses on the lexical content of the URLs, we can determine if our models are effective at recognising the patterns and keywords that typically appear in spam links. Fig. 3 shows the class distribution of the Spam URLs Classification dataset. The dataset is slightly imbalanced, with 31.9% of the URLs labelled as spam (True) and 68.1% labelled as not spam (False). This moderate imbalance highlights the importance of using evaluation metrics beyond accuracy.

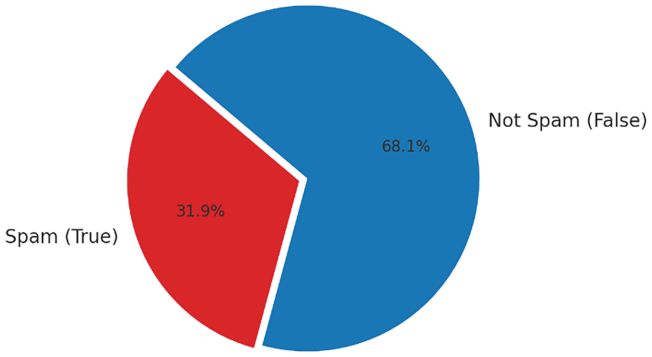

Figure 3: Class distribution in spam URLs classification dataset (spam vs. not spam)

3.1.2 Dataset 2: PhiUSIIL Phishing URL Dataset

The second dataset is the PhiUSIIL dataset [18], a well-known and widely used benchmark for phishing URL detection. It was obtained from the UCI Machine Learning Repository and contains a large number of examples, 235,795 instances in total. Each entry in this dataset includes 56 different features, all derived from analyzing the URL itself and the HTML source code of the associated webpage.

These features cover a broad range of characteristics, such as the length of the URL and domain, the rate at which certain characters appear, the number of subdomains, and whether the URL uses secure protocols like HTTPS. Because of its rich and detailed feature set, this dataset allows for a more advanced and heuristic-based analysis of phishing threats.

Fig. 4 shows the class distribution of the Spam URLs Classification dataset. The dataset is slightly imbalanced, with 57.2% of the URLs labelled as spam (True) and 42.8% labelled as not spam (False).

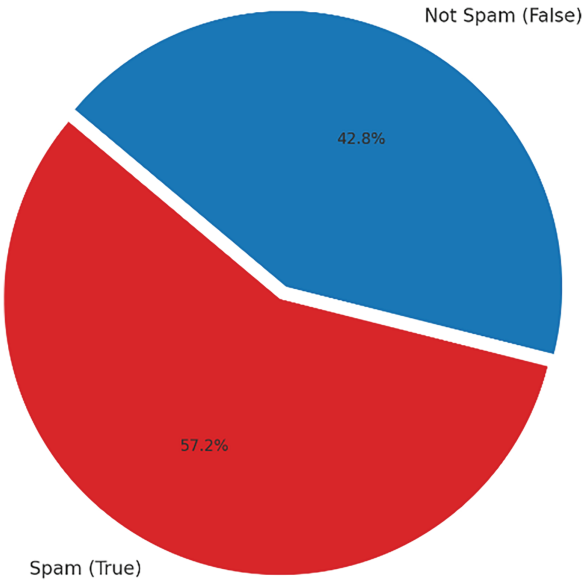

Figure 4: Class distribution in PhiUSIIL phishing URL dataset (spam vs. not spam)

By using both datasets, we get a solid foundation for testing our models under different conditions. One dataset challenges them with just the lexical patterns of a URL, while the other forces them to rely on a wide variety of technical and behavioural features.

3.2 Classification Models

To build and evaluate our detection system, we selected four powerful and widely respected classification algorithms. Each of these models is based on a different learning approach, which allowed us to compare a variety of techniques and understand how each one performs under different conditions.

3.2.1 Random Forest (RF)

Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their outputs to make the final prediction. This approach helps reduce overfitting and makes the model more stable and accurate. Random Forest works especially well with high-dimensional data and can also tell us which features are most important for making decisions. Because of its robustness and versatility,

we found it to be an excellent choice for both our datasets, whether we used simple lexical features or complex, structured ones [19].

3.2.2 Support Vector Machine (SVM)

SVM is a powerful classifier that aims to find the best possible boundary (called a hyperplane) that separates different classes in the data. It works well even when the data is not linearly separable, thanks to the use of kernel functions that transform the data into higher dimensions. Its strength lies in handling complex and subtle patterns, making it a solid option for cybersecurity tasks where threats are often cleverly disguised [20].

3.2.3 Naïve Bayes (NB)

Naïve Bayes is a probabilistic model based on Bayes' Theorem and assumes that all features are independent from one another, a simplification that often works surprisingly well in practice. It's especially fast and efficient, which makes it ideal for handling large datasets [21]. For Dataset 1, which includes text-based features from URLs, we used the Multinomial Naïve Bayes version, which is well-suited for this type of data. For Dataset 2, which includes continuous numerical features, we applied Gaussian Naïve Bayes.

3.2.4 Artificial Neural Network (ANN)

We used an Artificial Neural Network in the form of a Multi-Layer Perceptron (MLP), which consists of layers of interconnected nodes (neurons) that can learn complex, non-linear relationships in the data. ANNs are particularly powerful when dealing with problems where patterns are not obvious or straightforward [22]. This makes them a strong candidate for detecting more advanced and sophisticated malicious URLs that might slip past simpler models.

3.3 Model Updating Strategy

To ensure that the detection system remains effective against emerging phishing techniques, the models used in this study can be updated periodically through incremental retraining. Newly collected phishing URLs from threat-intelligence feeds, public repositories, or institutional logs can be added to the existing training dataset. Once new samples are incorporated, the preprocessing pipeline is re-applied to normalise and tokenise the URLs, and updated lexical and structural features are regenerated. Because the models employed in this work, Random Forest, SVM, Naïve Bayes, and ANN, have relatively low training overhead, they can be retrained efficiently on updated datasets without requiring extensive computational resources. This approach enables continuous improvement of the classifier, helping to maintain robustness against rapidly evolving phishing tactics.

3.4 Experimental Setup and Evaluation

To ensure fair and consistent testing, both datasets were split into two parts: 80% of the data was used for training the models, and the remaining 20% was set aside for testing. We used stratified splitting, a method that maintains the original proportion of malicious and benign URLs in both the training and testing sets. This helps prevent issues related to class imbalance and ensures that our models are evaluated on representative data.

To carefully assess how well each model performed, we used a combination of widely accepted evaluation metrics:

- *Accuracy*: This measures the overall correctness of the model by calculating the ratio of correct predictions to the total number of predictions. While it gives a general idea of performance, it may not always reflect performance on imbalanced datasets.
- *F1-Score*: The F1-score combines precision and recall into a single number using their harmonic mean. This metric is especially useful when the dataset is imbalanced, as it balances the trade-off between false positives and false negatives.
- *ROC-AUC Score*: This stands for "Receiver Operating Characteristic-Area Under the Curve". It gives an overall picture of the model's ability to distinguish between malicious and benign URLs across all possible thresholds. A higher ROC-AUC score indicates better discrimination between the classes.

In addition to these metrics, we also created confusion matrices for each experiment to get a more detailed view of the model's predictions. Each confusion matrix shows four key numbers:

- *True Positives (TP)*: Malicious URLs correctly identified as malicious.
- *True Negatives (TN)*: Legitimate URLs correctly identified as legitimate.
- *False Positives (FP)*: Legitimate URLs incorrectly identified as malicious.
- *False Negatives (FN)*: Malicious URLs that the model failed to detect.

These visualizations help us understand exactly where each model is getting things right or going wrong and offer valuable insights for improving future performance.

4 Results and Analysis

This section presents the empirical results obtained from applying our methodology. We detail the performance of the four selected models on both the Spam URL and PhiUSIIL Phishing URL datasets and defending our findings by comparing them against established benchmarks from prior research.

4.1 Performance on Spam URL Classification

This section presents the results of our first set of experiments, which were conducted on the Spam URLs Classification Dataset. In this case, we used TF-IDF vectorization to convert the raw URL strings into numerical features that could be processed by the models. The overall performance of the four selected classifiers, Random Forest, Support Vector Machine (SVM), Naïve Bayes, and Artificial Neural Network (ANN) is summarized in Table 2.

Table 2: Models overall performance comparison on Spam URLs Classification Dataset

Models	Accuracy (%)	F_Score (%)	ROC-AUC (%)
Random Forest (RF)	99.62	99.4	99.96
Support Vector Machine (SVM)	98.27	97.27	99.58
Naïve Bayes (NV)	93.53	89.94	97.26
Artificial Neural Network (ANN)	99.23	98.8	99.78

Across the board, the models performed very well, showing strong capability in distinguishing between spam and legitimate URLs. Among them, Random Forest stood out as the top performer, achieving an exceptionally high accuracy of 99.62% and an F1-Score of 99.40%. This result highlights the strength of ensemble learning in handling text-based features like those generated by TF-IDF.

The Artificial Neural Network also showed excellent performance, reaching an accuracy of 99.23%, indicating its strong ability to learn from complex patterns in the data. The SVM model followed closely with

an accuracy of 98.27%, proving to be a reliable choice as well. While the Naïve Bayes classifier achieved a comparatively lower accuracy of 93.53%, it still performed reasonably well given its simplicity and speed.

When we compare our results with previous work, the improvements are self-evident. For example, our Random Forest model trumped Jilani and Sultana's [11] own accuracy of 97.39% by more than 2.2 percentage points, a new benchmark. More impressive still is that our model outperformed Kotni et al. [13], who used a more advanced deep learning model, DistilBERT, but managed only 95.02% accuracy.

To better understand the performance of our models beyond overall accuracy, we closely examined the confusion matrices of each classifier on the Spam URL dataset, as presented in Fig. 5.

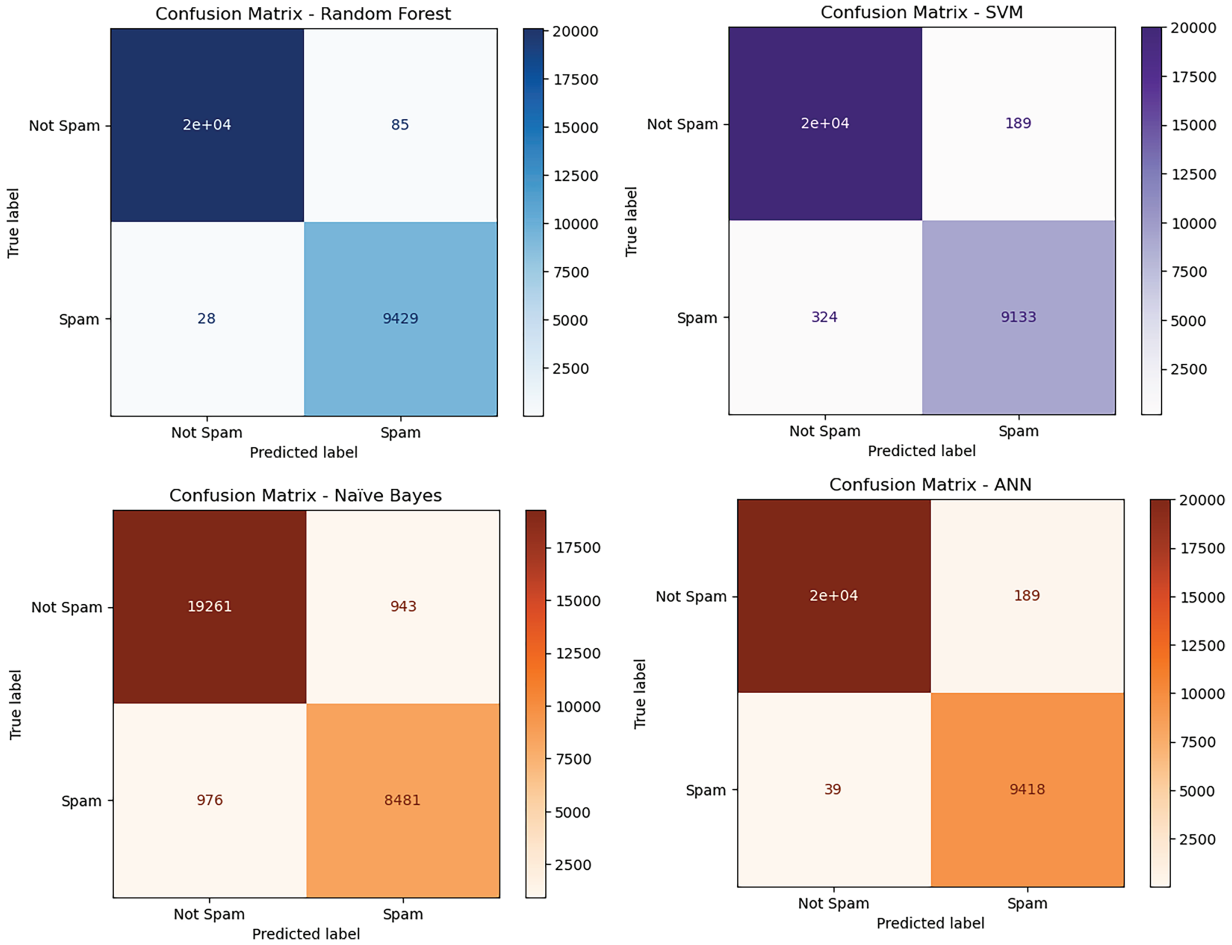

Figure 5: Visualisation of confusion matrices of different classifiers on spam URLs classification dataset

Results indicate that our fairly straightforward feature engineering strategy, together with a strong model such as Random Forest, is not merely highly effective but also computationally efficient. This makes it a very good option for real-time spam URL detection systems, where both speed and accuracy must be high simultaneously.

4.2 Performance on PhiUSIIL Phishing URL Detection

In the second set of experiments, we evaluated our models using the PhiUSIIL Phishing URL Dataset, which is known for its complexity and rich set of features. This dataset includes detailed heuristic and structural information, making it a strong benchmark for testing phishing detection models. The performance of each model on this task is summarized in Table 3.

Table 3: Models' overall performance comparison on PhiUSIIL phishing URL dataset

Models	Accuracy (%)	F_Score (%)	ROC-AUC (%)
Random Forest (RF)	99.99	99.99	99.99
Support Vector Machine (SVM)	99.97	99.97	99.99
Naïve Bayes (NV)	99.94	99.94	99.94
Artificial Neural Network (ANN)	99.98	99.99	99.99

All four models performed exceptionally well, with results that were very close to perfect. Once again, the Random Forest classifier achieved the highest accuracy of 99.99%, making only a tiny number of errors across a test set containing over 47,000 instances. This level of precision highlights the strength of the model in handling feature-rich, real-world data.

The other models also delivered outstanding results:

- The Artificial Neural Network (ANN) achieved an accuracy of 99.98%,
- Support Vector Machine (SVM) reached 99.97%, and
- Naïve Bayes followed closely with 99.94% accuracy.

These results place our models among the very best reported in the research community for this dataset. Our Random Forest and ANN models either match or surpass the top accuracies from previous studies. For instance, Siam et al. [24] achieved a similar performance using an LSTM-based deep learning model. Our models outperformed other best-performing methods as well. Our models surpassed those of Vajrobol et al. [6], achieving 99.97% accuracy, and Etem and Teke [5], with 99.7% accuracy using a Decision Tree model. A study by Mahdaouy et al. [23] evaluates a broad set of character-level and transformer-based models, including CharBiGRU, CharCNNBiLSTM, CySecBERT, SecureBERT, URLBERT, and DomURLs_BERT on the same PhiUSIIL dataset. The reported accuracies for these advanced architectures fall in the range of 99.78% to 99.82%, with DomURLs_BERT and URLBERT achieving approximately 99.80%–99.82%.

By marrying solid feature engineering with an effective model selection, we've managed to take phishing URL detection to the next level on one of the most challenging datasets to work with. This demonstrates not just how effective our method is, but also its potential in actual-world cybersecurity, where reliability and precision take centre stage.

Finally, to better understand the performance of our model beyond overall accuracy, we also carefully reviewed the confusion matrices for each classifier on the Spam URL dataset. Take a look at the breakdown in Fig. 6.

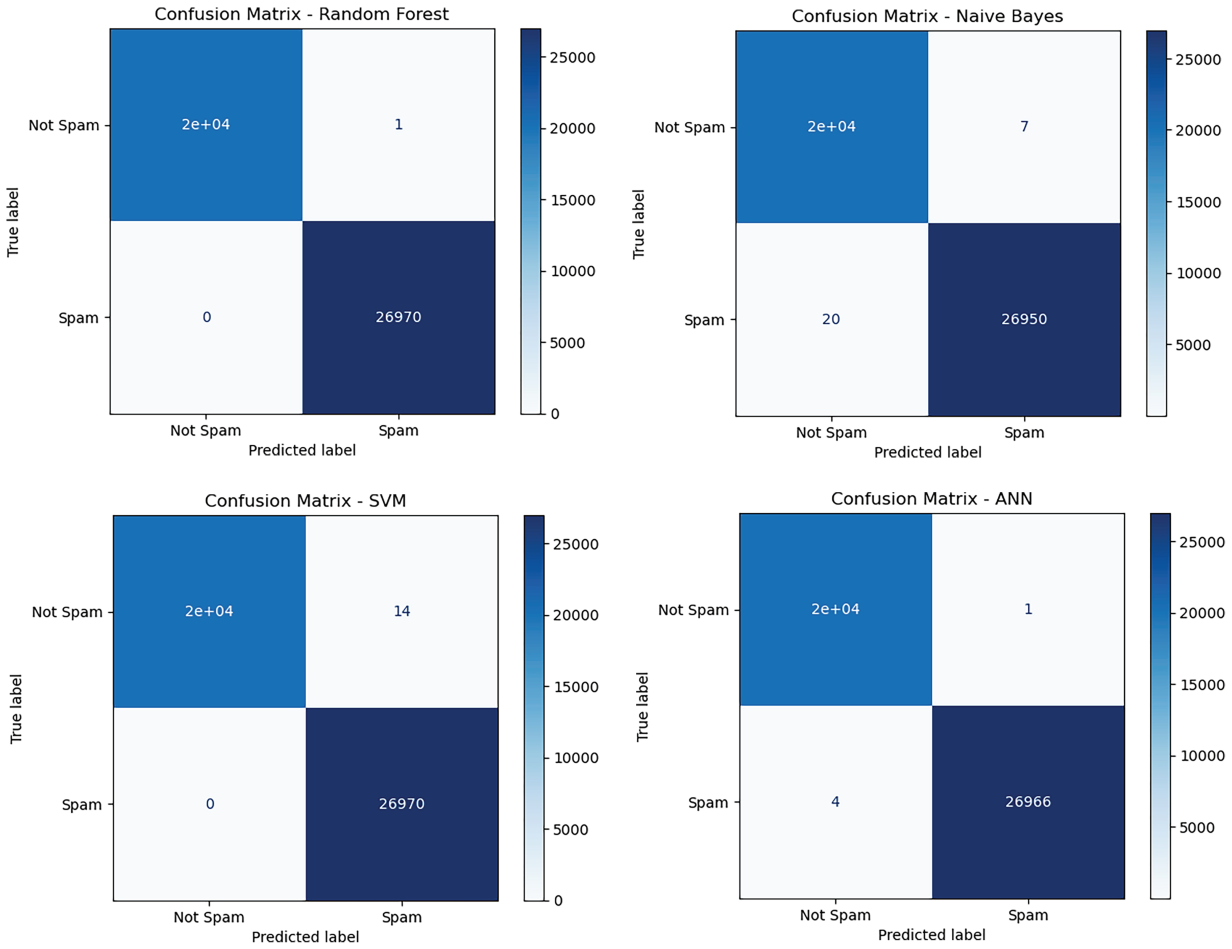

Figure 6: Visualization of confusion metrices of different classifiers on dataset 2

4.3 Discussion

The results from both experiments strongly support the effectiveness of our overall approach. Across both datasets, the Random Forest and Artificial Neural Network (ANN) models consistently stood out, delivering the highest performance. This reinforces their ability to handle the complex and often subtle patterns needed for accurate URL classification.

On Dataset 1 (the Spam URL dataset), our use of TF-IDF for statistical lexical analysis proved to be highly effective. Even without manually engineered features, this approach allowed the models to pick up on patterns and characteristics that are commonly found in spam URLs. These results show that with the right feature extraction method, models can learn a great deal simply from the structure and content of the URL text.

In contrast, the results from Dataset 2 (the PhiUSIIL phishing dataset) demonstrate the power of heuristic and structural feature engineering. By designing a comprehensive, high-dimensional feature set that captures a wide range of URL behaviors and characteristics, we enabled our models to achieve near-perfect accuracy. This suggests that when rich, meaningful features are available, machine learning models can perform exceptionally well, even in complex real-world scenarios like phishing detection.

Overall, our findings not only confirm the strengths of well-established models like Random Forest and ANN, but also set a new benchmark, particularly on the PhiUSIIL dataset, which is widely used for phishing detection research. This positions our work as a strong contribution to the ongoing development of reliable, high-accuracy cybersecurity solutions.

5 Conclusion

This study set out to evaluate and benchmark the performance of several widely used machine learning and neural network models for the important task of classifying malicious URLs. By using two carefully chosen datasets and applying different feature engineering strategies tailored to each, we have shown that it is possible to achieve state-of-the-art and near-perfect accuracy in detecting both spam and phishing URLs.

Our results clearly demonstrate the power of ensemble methods and neural networks in this domain. Among the models tested, Random Forest and Artificial Neural Networks (ANN) consistently delivered the best performance, whether the task was driven by lexical patterns, as in the spam detection dataset, or by a rich set of heuristic features, as in the phishing detection dataset.

The strong results on the spam dataset highlight the effectiveness of the TF-IDF vectorization approach, showing that statistical text analysis alone can uncover hidden patterns in URLs without the need for manual feature creation. Meanwhile, the exceptional 99.99% accuracy achieved on the PhiUSIIL phishing dataset confirms that our comprehensive, heuristic-based feature set successfully captures subtle signals that distinguish phishing attempts from legitimate links.

In summary, this work makes a meaningful contribution to the field of cybersecurity. Not only does it validate the reliability and effectiveness of well-known classification models, but it also sets a new performance benchmark, particularly for the challenging and widely used PhiUSIIL dataset. These results push the current boundary of what's achievable in malicious URL detection.

6 Future Scope

While this study sets a new benchmark in the accuracy of malicious URL classification, there are several promising directions for future research that can further enhance the robustness, transparency, and practicality of cybersecurity solutions.

6.1 Exploration of Advanced Models

The impressive results achieved by Random Forest and Artificial Neural Networks (ANN) form a strong baseline. However, future work could explore additional ensemble techniques like Gradient Boosting Machines (GBM) and XGBoost, which have consistently delivered top performance in many classification challenges. These models may offer further improvements in both accuracy and generalization. Additionally, more advanced deep learning models such as Long Short-Term Memory (LSTM) networks and Transformer-based architectures like BERT could be particularly valuable. These models are well-suited for capturing sequential and contextual patterns in URLs, which may help in detecting more sophisticated or obfuscated threats.

6.2 Hyperparameter Optimization and Advanced Feature Engineering

Although our models performed exceptionally well, there is room to fine-tune them for even better results. Future research can focus on automated hyperparameter tuning using techniques such as Grid Search, Random Search, or Bayesian Optimization to systematically identify the best configuration for each model. In parallel, deeper exploration of feature engineering could further enhance performance. This might

include creating interaction features, applying dimensionality reduction techniques, or using automated feature selection algorithms to isolate the most impactful predictors. These approaches could result in models that are not only more accurate but also faster and easier to deploy.

6.3 Scalability and Real-Time Application

While our current study evaluates performance on static datasets, the real-world deployment of these models would require them to perform efficiently in real-time and high-throughput environments. Future work should assess how well these models scale when applied to streaming data or significantly larger datasets. This includes benchmarking latency, memory usage, and throughput to ensure they can operate without bottlenecks in practical settings, such as network firewalls, email filtering systems, or web gateways. Adapting these models for online learning or incremental updates could also help maintain their effectiveness in the face of constantly evolving cyber threats.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Tran Minh Bao and Kumar Shashvat; methodology, Tran Minh Bao and Nguyen Gia Nhu; software, Kumar Shashvat and Dac-Nhuong Le; validation, Kumar Shashvat, Tran Minh Bao and Nguyen Gia Nhu; formal analysis, Tran Minh Bao, Nguyen Gia Nhu and Dac-Nhuong Le; investigation, Kumar Shashvat and Tran Minh Bao; resources, Tran Minh Bao and Kumar Shashvat; data curation, Kumar Shashvat and Tran Minh Bao; writing—original draft preparation, Kumar Shashvat, Tran Minh Bao and Nguyen Gia Nhu; writing—review and editing, Dac-Nhuong Le; visualization, Kumar Shashvat, Tran Minh Bao and Nguyen Gia Nhu; supervision, Nguyen Gia Nhu and Dac-Nhuong Le; project administration, Nguyen Gia Nhu and Dac-Nhuong Le; funding acquisition, Nguyen Gia Nhu and Dac-Nhuong Le. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Patra C, Giri D, Maitra T, Kundu B. A comparative study on detecting phishing URLs leveraging pre-trained BERT variants. In: Proceedings of the 2024 6th International Conference on Computational Intelligence and Networks (CINE); 2024 Dec 19–21; Bhubaneswar, India. p. 1–6. doi:10.1109/CINE63708.2024.10881521.
2. Mahajan R, Siddavatam I. Phishing website detection using machine learning algorithms. Int J Comput Appl. 1970;181(23):45–7. doi:10.5120/ijca2018918026.
3. Jalil S, Usman M, Fong A. Highly accurate phishing URL detection based on machine learning. J Ambient Intell Humaniz Comput. 2023;14(7):9233–51. doi:10.1007/s12652-022-04426-3.
4. Alsariera YA, Elijah AV, Balogun AO. Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations. Arab J Sci Eng. 2020;45(12):10459–70. doi:10.1007/s13369-020-04802-1.
5. Etem T, Teke M. Advanced phishing detection: leveraging t-SNE feature extraction and machine learning on a comprehensive URL dataset. Acta Infologica. 2024;8(2):213–21. doi:10.26650/acin.1521835.
6. Vajrobol V, Gupta BB, Gaurav A. Mutual information based logistic regression for phishing URL detection. Cyber Secur Appl. 2024;2:100044. doi:10.1016/j.csa.2024.100044.

7. Gałka W, Bazan JG, Bentkowska U, Mrukowicz M, Drygaś P, Ochab M, et al. Self-tuning framework to reduce the number of false positive instances using aggregation functions in ensemble classifier. Procedia Comput Sci. 2024;246:4028–37. doi:10.1016/j.procs.2024.09.241.
8. Ren F, Jiang Z, Liu J. A bi-directional LSTM model with attention for malicious URL detection. In: Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC); 2019 Dec 20–22; Chengdu, China. p. 300–5. doi:10.1109/IAEAC47372.2019.8997947.
9. Lakshmi L, Reddy MP, Santhaiah C, Reddy UJ. Smart phishing detection in web pages using supervised deep learning classification and optimization technique ADAM. Wirel Pers Commun. 2021;118(4):3549–64. doi:10.1007/s11277-021-08196-7.
10. Rawla A, Singh S, Daniyal M, Dubey P. Detection of phishing attacks in PhiUSIIL dataset using deep learning. Procedia Comput Sci. 2025;259:543–52. doi:10.1016/j.procs.2025.04.003.
11. Jilani AK, Sultana J. A random forest based approach to classify spam URLs data. In: Proceedings of the 2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS); 2022 Jun 22–23; Manama, Bahrain. p. 268–72. doi:10.1109/ICETSIS55481.2022.9888849.
12. Ubing AA, Kamilia S, Abdullah A, Jhanjhi NZ, Supramaniam M. Phishing website detection: an improved accuracy through feature selection and ensemble learning. Int J Adv Comput Sci Appl. 2019;10(1):252–7. doi:10.14569/ijacsa.2019.0100133.
13. Kotni S, Potala C, Sahoo L. Spam detection using deep learning models. Int J Adv Res Eng Technol. 2022;13(5):55–64. doi:10.17605/OSF.IO/NT4.
14. Jaswanthi T, Harshitha T, Tanya S, Belwal M. Malicious URL detection: comparative study of machine learning algorithms. In: Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT); 2024 Jun 24–28; Kamand, India. p. 1–5. doi:10.1109/ICCCNT61001.2024.10725561.
15. Sahingoz OK, BUBEr E, Kugu E. DEPHIDES: deep learning based phishing detection system. IEEE Access. 2024;12:8052–70. doi:10.21227/4098-8c60.
16. Zamir A, Khan HU, Iqbal T, Yousaf N, Aslam F, Anjum A, et al. Phishing web site detection using diverse machine learning algorithms. Electron Libr. 2020;38(1):65–80. doi:10.1108/el-05-2019-0118.
17. Shivam B. Spam URL prediction [Dataset]; 2017 [cited 2025 Oct 1]. Available from: https://www.kaggle.com/datasets/shivamb/spam-url-prediction.
18. Prasad A, Chandra S. PhiUSIIL: a diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Comput Secur. 2024;136:103545. doi:10.1016/j.cose.2023.103545.
19. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. doi:10.1023/a:1010933404324.
20. Steinwart I, Christmann A. Support vector machines. New York, NY, USA: Springer; 2008. doi:10.1007/978-0-387-77242-4.
21. Rish I. An empirical study of the Naive Bayes classifier. IJCAI, 2001 Workshop Empir Methods Artif Intell. 2001;3(22):41–6.
22. Agatonovic-Kustrin S, Beresford R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J Pharm Biomed Anal. 2000;22(5):717–27. doi:10.1016/S0731-7085(99)00272-1.
23. Mahdaouy AE, Lamsiyah S, Idrissi MJ, Alami H, Yartaoui Z, Berrada I. DomURLs_BERT: pre-trained BERT-based model for malicious domains and URLs detection and classification. arXiv:2409.09143. 2024.
24. Siam MNH, Hallaji E, Razavi-Far R. Enhancing cyberspace security with phishing detection and defense using machine learning models. In: Proceedings of the 2025 13th International Symposium on Digital Forensics and Security (ISDFS); 2025 Apr 24–25; Boston, MA, USA. p. 1–6. doi:10.1109/ISDFS65363.2025.11012088.
25. Tian Y, Yu Y, Sun J, Wang Y. From past to present: a survey of malicious URL detection techniques, datasets and code repositories. arXiv:2504.16449. 2025.