Combatting Phishing Attacks: Leveraging Machine Learning for Real-Time Detection in Penetration Testing

Ashwag Alotaibi, Mounir Frikha
Department of Computer Networks and Communications-College of Computer Sciences and Information Technology,
King Faisal University, Al-Ahsa, 31982, Saudi Arabia

***Abstract*—Phishing attacks continue to pose a significant threat to individuals and organizations, driven by the increasing sophistication of cybercriminal techniques and the rapid expansion of digital services. Traditional detection approaches, such as blacklist-based and rule-based systems, are often ineffective against newly generated or obfuscated phishing URLs. This study proposes a machine learning (ML)-based framework intended for integration within penetration testing environments. The approach leverages multiple supervised learning algorithms, including Random Forest (RF), Support Vector Machine (SVM), and XGBoost, trained and evaluated using the PhiUSIIL Phishing URL Dataset, a large-scale benchmark dataset containing phishing and legitimate URL samples. A comprehensive preprocessing pipeline and feature engineering strategy are employed to enhance model performance. Experimental results demonstrate exceptionally high detection accuracy, with RF and XGBoost achieving near-perfect classification performance across key evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The proposed system is further designed for real-time deployment, enabling integration into penetration testing workflows for proactive security assessment. Despite promising results, limitations related to dataset characteristics and real-world generalization are acknowledged. Overall, this research highlights the effectiveness and practical applicability of ML-based approaches in strengthening phishing detection and advancing modern cybersecurity defences.**

***Keywords*—Phishing detection; machine learning; real-time detection; penetration testing; URL analysis**

I. Introduction

A. Background

Phishing attacks have emerged as one of the most prevalent and dangerous forms of cyberattacks on individuals, organisations, and the critical infrastructure spread throughout the world in recent years. The attacks are normally in fraudulent messages or rogue websites that are established to swindle the user into disclosing their sensitive information in terms of logins, financial, and personal information. Phishing is a dynamic and ongoing problem in cybersecurity because the attack surface has increased manyfold with the pace of digital transformation, internet banking, and cloud-based services [1].

Conventional phishing detection models, including blacklisting models and rule-based filters, have not been found to be effective in dealing with contemporary methods of attack. Such traditional approaches can still not detect newly created or obfuscated phishing URLs, which leads to a high level of false negatives [2]. Also, more and more attackers use automation and artificial intelligence to create advanced phishing campaigns that are very similar to authentic websites and thus more difficult to detect [3].

In order to overcome these limitations, ML technologies have become a popular subject to fill the gap in phishing detection. ML models can handle a lot of data and can identify the subtle patterns and project the past-observed attacks to new ones in real-time. The recent research has shown that supervised learning algorithms, like Random Forest and Support Vector Machines, are effective in obtaining a high detection accuracy [4]. As a result, incorporating the use of ML-based detection systems into the cybersecurity systems has emerged as an encouraging route toward an improved phishing defence infrastructure.

B. Problem Statement and Motivation

Although machine learning-based phishing detection has been developed, a number of challenges have not been addressed. The majority of existing literature is interested in offline analysis, where models are trained and are tested on fixed datasets without reference to real-time deployment conditions [5]. This weakness makes them less applicable in dynamic environments where immediate detection and response are essential, like in the case of penetration testing.

Moreover, most of the existing methods fail to incorporate phishing detection solutions into the penetration testing processes. Penetration testing is a proactive security measure that recreates real-world attacks to detect vulnerabilities, but it usually does not use intelligent and automated tools in detecting phishing attacks during testing processes. Moreover, other concerns, like a high false-positive rate and high computational overhead, do not allow using ML models in real-time systems [6].

This study is thus driven by the need to come up with an effective and realistic machine learning-based system that would be efficient in identifying phishing attacks in real time, that is, in a penetration testing context. The proposed research will fill this gap by focusing on these limitations to combine theoretical ML models and their practical implementation in cybersecurity.

C. Objectives and Contributions

The main aim of the proposed research is to develop and evaluate an ML-based phishing detection framework for penetration testing environments and assess its potential for future

real-time deployment. To do this, the following objectives need to be reached in the study:

- To train and evaluate a phishing model based on benchmark datasets of phishing.
- To implement and compare multiple ML algorithms, including RF, SVM, and XGBoost.
- To create a framework that can perform real-time phishing detection.
- To measure the performance of the models by applying conventional metrics like accuracy, precision, recall, and F1-score.
- To determine the viability of the ML-based detection integration into the penetration testing procedures.

The study is of value to the research, since it proposes a valuable and scalable phishing detection system that would provide a high accuracy rate and can be implemented in real-time.

D. Paper Organization

Below is the study structure:

- Section II includes an extensive overview of the recent literature on the topic of phishing detection by the use of machine learning methods and the critical analysis of the current methods and gaps in the literature.
- Section III explains the intended methodology, which incorporates gathering of data, preprocessing, selection of features, and development of models.
- Section IV will be the experimental design and the evaluation process, together with the results and performance analysis of the applied models.
- Section V involves the discussion of the findings, their implications, and limitations.
- Section VI concludes the study and provides recommendations on how future research might be conducted.

II. Related Work

A. Overview of Existing Works

The past few years have seen an extensive amount of studies on phishing detection, especially in the context of the application of ML and DL methods. Malicious activities have been studied by researchers through an analysis of URL features, webpage contents, and behavioral patterns with maximum accuracy.

The early modern methods (after 2020) focus on the use of supervised ML models that are trained using structured datasets. As an example, the study [2] suggested a character-level convolutional neural network (CNN) to detect phishing URLs that showed a good level of performance in detecting obfuscated URLs. On the same note, the study [4] came up with a hybrid deep learning model that integrates CNN and recurrent neural networks (RNN) and demonstrates better classification accuracy than the conventional ones.

Ensemble learning methods have been investigated in a number of studies in order to improve detection performance. In a thorough study of the various ML algorithms, [7] revealed that the Random Forest and Gradient Boosting are better than single classifiers in the phishing detection processes. Similarly, the study [3] also focused on the efficiency of ensemble models like XGBoost in working with large datasets of malicious URLs.

The feature-based methods are quite popular because they are interpretable and efficient. In [6], the authors employed both lexical and host-based attributes, including the URL length and the domain age, as input to train classifiers using ML, with high detection rates, but with comparatively low computational cost. Equally, [8] concentrated on URL-related aspects to identify phishing sites without webpage content, which is why the method is appropriate for real-time systems.

DL models are also on the increase due to the automatic extraction of features in raw data. The authors of [9] used long short-term memory (LSTM) networks in a sequential URL analysis, and it is seen that it has a better result in finding advanced phishing patterns. Another study [10] suggested a CNN-based model that takes the URL strings as images and gives competitive results.

The current studies have devoted more attention to the issues of real-time detection and deployment. In [11], the authors developed a light-weight ML model that is optimized to work with real-time phishing detection and minimizes the use of computational power, and still provides the same accuracy.

The hybrid method, combining various methods, has also proved to be promising. The study [12] combined feature engineering and ensemble learning and attained better rates of detection. Also, [4] suggested a hybrid ML-DL system that uses both structured and unstructured data to detect phishing. Moreover, explainable artificial intelligence (XAI) is also introduced to enhance the transparency of the models. The use of feature importance methods to explain the predictions of the ML was applied to [13], which allowed a more accurate perception of phishing indicators. This is more so in the case of cybersecurity applications, where transparency in decision is of utmost importance.

Fig. 1 presents a classification of the existing literature on phishing detection, highlighting the different machine learning, deep learning, and hybrid approaches.

The literature also covers adversarial resistance and changing phishing strategies in recent research. [14] investigated the use of URLs by attackers to bypass detection models, and it was found that adaptive ML systems are necessary. On the same note, [15] also surveyed phishing detection methods and the need to continually update the model.

In spite of such developments, the majority of the studies are based on offline datasets and are not coupled with penetration testing frameworks. Although accuracy gains can be noticed, the feasibility of cybersecurity in a real-world setting is not yet broadly applicable.

Table I provides a comparative analysis of existing studies on phishing detection, summarizing their methodologies, datasets, key features, performance, and identified limitations.

TABLE I. ANALYSIS OF RELATED WORK

Author	Year	Method	Dataset	Key Features	Accuracy	Limitation
[16]	2024	Hybrid ML (RF + GB + SVM)	Mixed phishing datasets	URL + content hybrid features	High	Feature selection dependency
[17].	2023	CNN, LSTM, LSTM-CNN	Phishing URL dataset	Deep feature fusion	Very High	High training complexity
[18]	2023	Hybrid DNN–LSTM	URL dataset	Sequential URL patterns	High	High training time
[4]	2023	Hybrid ML (URL-based)	Mixed URL datasets	Multi-source URL features	High	URL-only, no content analysis
[19]	2021	CNN + Multi-head Attention	Imbalanced URL dataset	Attention-based URL encoding	High	Imbalanced class sensitivity
[2]	2020	CNN	PhishTank + Alexa URLs	Character-level URL features	High	Computational cost
[11]	2020	Real-time ML (SVM, RF)	PhishTank URL dataset	Optimised URL features	High (97.32%)	Limited to URL-only analysis
[12]	2020	AdaBoost + MultiBoosting	UCI phishing dataset	Combined URL + content features	High	Limited generalisation
[14]	2020	Adversarial ML (GAN-based)	URL adversarial dataset	Adversarial-robust URL features	High	Computationally expensive
[13]	2020	ML benchmarking	Large phishing benchmark	Comprehensive feature evaluation	High	Dataset-specific findings
[20]	2020	CNN + HTML analysis	URL + HTML phishing dataset	Raw URL + HTML characteristics	High	Content crawling required
[6]	2019	Random Forest, NLP	Custom URL dataset (73,575 URLs)	NLP + lexical URL features	Very High (97.98%)	Language-dependent features
[15]	2019	Hybrid Ensemble Feature Selection	Phishing websites dataset	CDF-g optimised URL features	High	Dataset-specific performance
[8]	2019	SVM, RF	URL dataset	Lexical + host-based features	Very High (99.31%)	Relies on third-party services
[7]	2019	CNN + LSTM	URL + image + text dataset	Multi-modal combined features	High (93.28%)	High computational cost

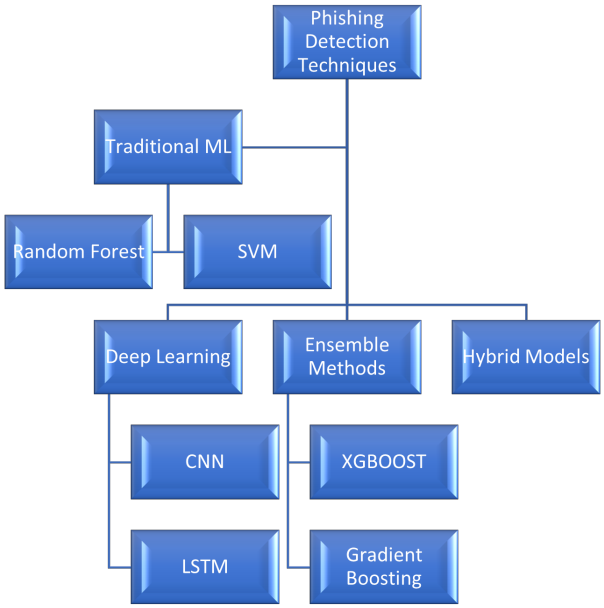

Fig. 1. Literature classification

Recent studies on ML and DL techniques of phishing detection have been synthesized. One can mention that most of the methods are highly precise, but most of them are limited by the aspects of computational complexity, the lack of real-time functionality, and the inability to react to dynamic conditions.

Recent phishing detection research has shifted toward transformers, graph neural networks (GNNs), and pretrained language models, improving detection of complex phishing attacks through better content and relationship analysis, and marking a move from traditional feature-based methods to representation-learning frameworks.

B. Critical Analysis

The analyzed articles reflect significant advances in phishing detection using ML and DL models. Traditional ML models, such as RF and SVM, are easy to use, interpretable, and efficient [6]. They are particularly used in real-time applications as their computational needs are also not very high.

On the other hand, DL models, such as CNN and LSTM models, are more accurate, as they extract complex features of raw data automatically [9][10]. Nevertheless, such models usually demand a lot of computer processing and more training time, and are not so useful in real-time.

Ensemble predictors, including XGBoost and Gradient Boosting, have the best performance and efficiency trade-offs, which makes them the best solution to the phishing detection system [3]. Hybrid methods also increase detection capabilities, since they use a combination of various techniques, but they also add complexity [4].

One severe drawback of most studies is that they use stagnant data and offline assessment. There are limited studies that address the deployment issues, including latency, scalability, and integration with the current cybersecurity tools. Also, most of the models are not explainable, which is a key to trust and acceptance in real-world systems [13].

C. Research Gap and Novelty

The phishing detection literature still has some gaps despite the fact that a lot of research has been conducted in this area. To begin with, most of the current methods are centered on offline detection and are not concerned with the issue of real-time implementation. This limits their application in dynamic systems such as penetration testing, where an instant response is required.

Second, the application of machine learning-based phishing detection is not well integrated into the context of penetration

testing. The majority of the research considers phishing detection as an independent entity instead of a part of a larger cybersecurity plan.

Thirdly, the current models tend to focus on the accuracy and not on the speed of detection and the efficiency of the computation, which are essential in real-time systems. Also, the inability to explain most ML models diminishes their application in security processes.

To address these gaps, the present study introduces a real-time machine-learning-based phishing detector implemented within the framework of penetration testing. The work focuses on the accuracy of the detection and the efficiency of the operation, which offers a valid solution when it comes to the practice of cybersecurity in reality.

III. Methodology

A. System Overview

The proposed study suggests a machine learning-driven phishing detector in a real-time environment to be used in the context of a penetration testing approach. The system accepts input URLs or web-based attributes, derives useful features, and classifies them as being phishing or legitimate with trained ML models.

The general process flow of the proposed system will include five key steps, namely, data collection, preprocessing, feature extraction and selection, model training, and real-time prediction. These phases are aimed at ensuring high detection rates, as well as effective implementation, which is a key ingredient of real-time cybersecurity applications [3].

Fig. 2 illustrates the overall architecture of the proposed phishing detection system.

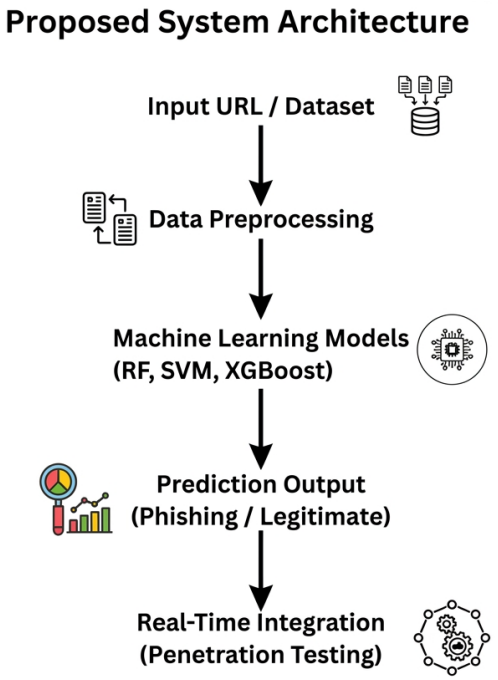

Fig. 2. Proposed system architecture.

B. Data Collection

The primary dataset used in this study is the PhiUSIIL Phishing URL Dataset, which contains a large collection of phishing and legitimate URLs along with URL-based, domain-based, security-related, and content-related features. The dataset was selected because of its size, diversity, and suitability for training and evaluating machine learning models for phishing detection.

The UCI Phishing Websites Dataset and PhishTank repository were initially reviewed during the literature survey and methodology design stages to identify commonly used phishing features and benchmark datasets reported in previous studies [6]. However, they were not used in the final experimental evaluation.

To ensure consistency and reproducibility, all experiments reported in Section IV were conducted exclusively using the PhiUSIIL Phishing URL Dataset. The dataset provides a comprehensive set of features to evaluate ML algorithms within a unified experimental framework.

C. Data Preprocessing and Cleaning

The collected data are preprocessed in a number of steps to enhance the quality of the data and to obtain the best performance of the ML models. Preprocessing of data is a very important phase, because raw data are usually full of noise, missing values, and inconsistency, which adversely affect the learning algorithms [2].

The preprocessing procedures that shall be adopted in the current study are:

- Handling Missing Values: Any records that have not been filled are dropped or filled in with any of the appropriate statistical techniques.
- Data Cleaning: Duplicate entries and irrelevant features are eliminated to reduce redundancy.
- Feature Encoding: Categorical features are converted into numerical representations using encoding techniques such as label encoding.
- Normalization: The values of features are brought into the standard range so that there is uniformity among the variables, especially in models that are sensitive to feature size, like SVM.

These steps are used to ensure that the data is clean, consistent, and can be used to train the model effectively.

D. Feature Extraction and Selection

Feature engineering is a concern in phishing detection because the quality of input features is directly proportional to model performance. The dataset features represent URL characteristics, domain information, security indicators, content attributes, and behavioural properties. For the experimental evaluation, all numerical features provided in the dataset were initially retained after preprocessing. The six features listed below are presented as representative examples of commonly

used phishing indicators that are frequently reported in the literature and are also reflected within the broader feature set of the PhiUSIIL dataset [8]:

- URL length
- Presence of HTTPS protocol
- Number of subdomains
- Use of special characters
- Domain age
- Redirection behaviour

In order to enhance the efficiency of a model and to decrease computational cost, the application of feature selection methods is used to determine the most important attributes. It is possible to reduce redundant or less informative features with the help of methods like correlation analysis and the rank of feature importance (e.g., by a Random Forest) [13].

E. Model Training and Configuration

This study chooses three ML algorithms to detect phishing. The selection of these models is based on their efficiency in classification activities and their common application in cybersecurity practices.

- RF: This is an ensemble learning algorithm that builds multiple decision trees and averages their predictions. RF has been characterized as having high strength and the capability to deal with high-dimensional data.
- SVM: This is a supervised learning system that determines the best hyperplane to use in making classification. SVMs can be useful in dealing with irregular decision boundaries and high-dimensional feature spaces.
- XGBoost: This is an improved ensemble boosting algorithm, which advances itself using gradient boosting. It is very effective, and it has been reported to be better in phishing detection tasks.

All the models are trained with a supervised learning method, whereby labelled data is separated into training and test sets, which is usually 80:20. RF and XGBoost use hyperparameters like tree depth, and SVM uses hyperparameters like kernel type to optimize performance.

F. Real-Time Detection Framework

This study proposes a conceptual framework for integrating trained machine learning models into a real-time phishing detection system that could be used in penetration testing environments. Unlike the traditional offline systems, the intended framework will operate in the dynamic mode and provide instant classification results to the incoming URLs.

Fig. 3 depicts the workflow of the real-time phishing detection framework, demonstrating how incoming URLs are processed, analyzed, and classified within a penetration testing environment.

The system can be implemented in either the form of a lightweight application or API, which allows penetration testers to test phishing attacks during attack simulation. This will encourage the practical value of the model and bridge the gap between the theoretical research and real practices of cybersecurity.

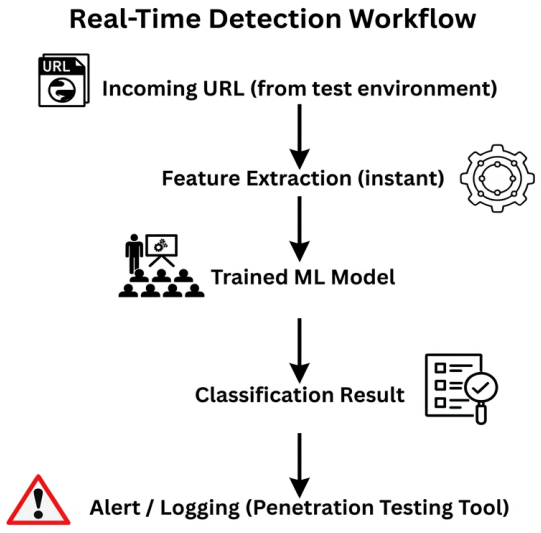

Fig. 3. Real-time detection workflow.

IV. Experimental Setup and Evaluation

A. Experimental Steps

The experimental analysis was performed in order to verify the possibilities of the ML models to identify phishing attacks on a massive dataset. It is developed using Python and the assistance of such packages as Scikit-learn, Pandas, NumPy, and XGBoost.

The process of the experiment was designed in the following way:

- Data Collection: Achieved by using PhiUSIIL Phishing URL Dataset with an exhaustive collection of URL, domain and content-based features.
- Data Preprocessing and Cleaning: The non-numeric data, such as the URL strings, the domain names and titles, were discarded to make them compatible with the machine learning algorithms. Missing values have been filled by zero imputation, and the dataset has been consistent as well.
- Feature Selection: Such features of the URL as security measures, behavioural characteristics and relevant numerical characteristics were employed.
- Dataset Splitting: The data was divided into training and testing groups in the proportion of 80:20 to come up with an objective evaluation.
- Model Training and Configuration: The processed data were used to train three machine learning models, which are: the RF, the SVM, and the XGBoost.
- Model Evaluation: The evaluation metrics of performance were based on standard evaluation metrics,

such as accuracy, precision, recall, F1-score, and ROC-AUC.

- Real-Time Simulation: The trained models were tested on how they could be integrated into real-time phishing detection systems in the testing penetrate.

B. Dataset Description

The dataset employed in this study is the PhiUSIIL Phishing URL Dataset, which consists of 235,795 examples and 56 features. The data has a wide variety of characteristics, including:

- URL-based features: URL length, character distribution, similarity index
- Domain-based features: Domain length, subdomains, TLD legitimacy probability
- Security features: HTTPS usage, obfuscation indicators
- Content features: Presence of forms, scripts, images, and external references
- Behavioural features: Redirections, pop-ups, and iframe usage

The data is categorized into two groups:

- 1 = Phishing
- 0 = Legitimate

The class distribution is as follows (see Table II):

TABLE II. CLASS DISTRIBUTION OF DATA

Class	Count
Phishing (1)	134,850
Legitimate (0)	100,945

Preliminary experiments with Synthetic Minority Oversampling Technique (SMOTE) were performed, but did not provide significant performance improvements, therefore, weighted learning was retained due to its lower computational overhead.

C. Evaluation Metrics

The following measures were used to evaluate the models (see Table III):

- Accuracy: Overall correctness of predictions
- Precision: Ability to avoid false positives
- Recall: Ability to detect phishing instances
- F1-score: Balance between precision and recall
- ROC-AUC: Overall classification capability across thresholds

TABLE III. EVALUATION METRICS

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	0.9999	0.9999	1.0000	1.0000	1.0000
SVM	0.9998	0.9997	1.0000	0.9998	1.0000
XGBoost	0.9999	0.9999	1.0000	0.9999	1.0000

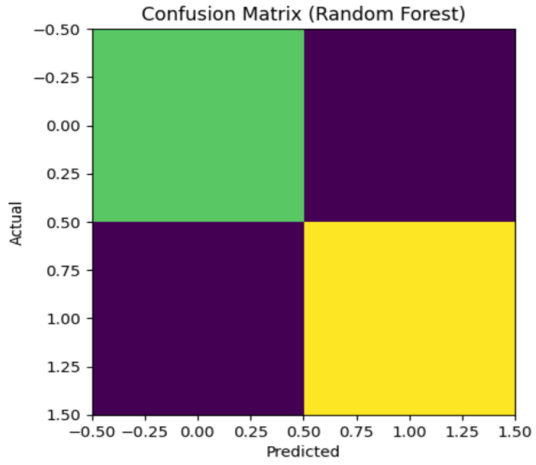

Fig. 4. Confusion matrix (Random Forest)

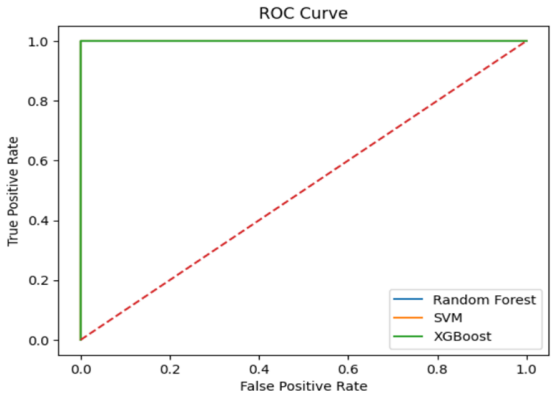

Fig. 5. ROC curve

D. Results and Analysis

The experimental findings has shown that the performance was outstandingly high in all the ML models that were tested. The model with the best performance was the RF, which implies that it made the correct classification of all the occurrences in the test dataset.

As illustrated in Fig. 4, the confusion matrix demonstrates the model's ability to accurately identify both classes within the evaluation dataset.

XGBoost also showed close to perfect performance with an accuracy of 0.9999, whereas the SVM showed a slightly lower, though very competitive performance with an accuracy of 0.9998. These results indicate that ensemble-based models (RF and XGBoost in particular) are very effective in identifying phishing attacks.

The confusion matrix also validates the strength of the RF model, as there are no false positives and false negatives, thus

indicating perfect classification. The same happens to the ROC curves of all models that tend to converge towards the ideal point, and the scores of AUC are 1.0, which proves to be very effective in differentiating between the classes of phishing and legitimate.

Fig. 5 further confirms the robustness of the evaluated models. The ROC curves remain close to the upper-left corner of the plot, indicating excellent discrimination capability between phishing and legitimate URLs across different classification thresholds.

However, this near-perfect performance may be an indicator of the potential limitations regarding the nature of the datasets or generalization of the models. The accuracy can be attributed to the fact that the separability of features in the dataset is very high, and the samples of phishing and legitimate websites are very different. Although this will improve the model performance in the controlled experiments, this may not necessarily be the case in the real world, where data is much noisier and more complex.

V. Discussion

A. Interpretation of Results

This is supported by the experimental results, which shows that ML models can be effective in recognizing phishing attacks if they are trained on a large and rich dataset. Although RF achieved the highest average performance among the evaluated models, the performance differences between RF and XGBoost were extremely small and statistically insignificant. Therefore, the superiority of RF should not be interpreted as conclusive. Instead, both ensemble-based models demonstrated comparable classification capability. This suggests that model selection for practical deployment should consider not only predictive accuracy but also operational factors such as inference latency, memory usage, scalability, and maintainability.

It is possible to explain the high effectiveness of the Random Forest by its ensemble character, i.e., several decision trees working together to enhance the level of predictions and decrease overfitting. This model is an effective representation of a webpage's behaviour in relation to a complex association between other features like URL structure, domain characteristics, and webpage behaviour. Equally, XGBoost is very effective because of its gradient boosting algorithm, which continuously enhances the model accuracy by emphasizing the misclassified cases [3].

The fact that it is almost impossible to recall all models is especially important regarding the area of cybersecurity. Good recall means that phishing is properly detected and the threat is reduced. This would be vital in a penetration testing situation where a phishing attack may pass by without realising a severe vulnerability. Regarding the real-time viewpoint, the findings indicate that the RF and the XGBoost would be very appropriate to be implemented in a penetration testing environment. The models not only offer a high level of accuracy but also offer efficient prediction time, which means that malicious URLs are detected quickly. This is in line to establish an effective, real-time phishing detection model, which can be used by cybersecurity experts when simulating an attack.

Additionally, the findings indicate the significance of feature engineering in phishing detection. The dataset employed in this research comprises a very diverse variety of characteristics, including URL-based characteristics, domain-based characteristics, and content-based characteristics, which play an important role in the model performance. This implies that the quality and diversification of features are important in enhancing the accuracy of detection.

B. Limitations

Although the performance is high in this study, a number of limitations should be acknowledged. To begin with, the almost flawless accuracy of all the models can suggest that the dataset is composed of very separable features, and the classification process can be made much easier. Although this contributes to great performance with controlled experiments, this may not be representative of the complexity of the real-world phishing, whereby attackers have continued to develop various methods of carrying out their activities in order to avoid detection.

Secondly, the models have been tested on a static dataset, which fails to take into consideration dynamic alterations in phishing behaviour over time. In practice, phishing attacks can take on new formats and unheard-of characteristics, and this can compromise the model's performance unless it is constantly updated [7].

The other one is the possible overfitting risk, especially in the case of ensemble models like Random Forest. These models are intended to be generalized, though, very high accuracy can reflect that the model has acquired dataset-specific trends, rather than overall phishing features. The study also pays more attention to URL and content-based properties, and it does not use contextual information (e.g., user behaviour or email metadata). The features might also be included in order to increase the detection capabilities and offer a more comprehensive approach to phishing detection.

An additional practical limitation concerns long-term model maintenance in production environments. Phishing strategies evolve continuously, creating concept drift that can reduce model effectiveness over time. Therefore, production deployment requires continuous monitoring, periodic retraining using recent threat intelligence, drift detection mechanisms, secure model update pipelines, and performance alerting. Without such operational controls, even highly accurate models may degrade under evolving attack patterns.

Lastly, even though the proposed system has great potential to be implemented in real-time, the execution of a complete real-time system was not carried out. The network latency, system integration, and scalability are aspects that need further research in order to be practical in case they need to be deployed in the penetration testing set-ups.

VI. Conclusion

A. Summary of Findings

This study examined how ML can be used to identify phishing attacks and, more specifically, how this can be applicable in real-time settings in penetration testing. The PhiUSIIL Phishing URL Dataset has been used to analyse a

comprehensive set of features, including URL-based, domain-based and content-based features.

The experimental results indicated that ML systems are capable of excelling in phishing. The evaluation indicated that Random Forest was the best, and its accuracy in the classification is perfect, which was also exhibited by XGBoost and SVM. These findings indicate the capability of ensemble-based models to identify complex tendencies of phishing attacks.

It was also necessary to make sure that the recalls were high in all the models and the phishing cases were properly identified during the test. It particularly applies to the sphere of cybersecurity, in which the failure to detect any malicious presence can have immense consequences. Overall, the results can support the utility of ML as a plausible instrument in phishing detection.

B. Core Contributions

The study contributes to the research area of cybersecurity and phishing detection in the following ways:

- It presents an ML-based phishing detector based on a large-scale and feature-sensitive dataset.
- It provides a comparative analysis of multiple ML models, including RF, SVM, and XGBoost.
- It also shows that high detection accuracy and efficiency can be attained to be used in real-time use.
- It also emphasises the potential to integrate machine learning models in the workflow of penetration testing.

However, compared to the rest of the studies performed, the current study focuses on performance and applicability, thereby establishing a gap between the theory and practice of cybersecurity implementation.

C. Real-World Value and Applicability

The study outcomes of the research have immense implications for the actual cybersecurity systems in the real-world. Very accurate and efficient models used in the evaluation of the models meant that they could be effectively applied in security tools to be applied in the detection of phishing attacks in real-time.

Specifically, through these models, penetration testers can detect phishing vulnerabilities in simulation attack mode to augment the process of complete security testing.

Additionally, these machine learning-based systems can be adopted by organizations to enhance protection against phishing attacks and minimize data breaches and financial losses. Such models are practical units of the modern cybersecurity systems due to their capability of detecting phishing attacks within a short period of time and at a high rate of success.

D. Future Work

Although the outcomes of this research are promising, there are a number of directions in which this research can be conducted to ensure that the phishing detection systems become even more efficient. First, DL models can be improved by more complex and dynamic trends of phishing using more sophisticated DL models, including LSTM networks and transformer-based networks. Second, complete functionality of a real-time detection system, i.e., a web-based API or a browser extension, would render the specified method of work even more pragmatically viable at an even larger scale. Besides, the active and constantly updated datasets, such as the data streams, would be involved to enable the models to evolve with the modifications of the phishing tactics upon their emergence. In addition to that, the feature sets may be expanded to incorporate the user behavioural evidence and email metadata that would offer a more sensible and holistic detection mechanism. Lastly, XAI would be provided to improve the transparency of the model and its credibility and practicability in the practice of cybersecurity in the real-life setting. In all, this would lead to more robust, larger, and more natural ML-based phishing detection systems.

FUNDING

This study was funded by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [GRANT No. KFU263466].

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [GRANT No. KFU263466]. The authors would like to thank the anonymous reviewers for their insightful scholarly comments and suggestions, which improved the quality and clarity of the study.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] R. Verma and A. Das, "What's in a url: Fast feature extraction and malicious url detection," in *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, 2017, pp. 55–63.

[2] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from url," *Electronics*, vol. 9, no. 9, p. 1514, 2020.

[3] D. Sahoo, C. Liu, and S. C. Hoi, "Malicious url detection using machine learning: A survey," *arXiv preprint arXiv:1701.07179*, 2017.

[4] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on url," *Ieee Access*, vol. 11, pp. 36 805–36 822, 2023.

[5] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672–694, 2021.

[6] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[7] M. A. Adebowale, K. T. Lwin, E. Sanchez, and M. A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text," *Expert Systems with Applications*, vol. 115, pp. 300–313, 2019.

[8] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and applications*, vol. 31, no. 8, pp. 3851–3873, 2019.

[9] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *Ieee Access*, vol. 10, pp. 36 429–36 463, 2022.

[10] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE access*, vol. 7, pp. 15 196–15 209, 2019.

[11] M. Sameen, K. Han, and S. O. Hwang, "Phishhaven—an efficient real-time ai phishing urls detection system," *Ieee Access*, vol. 8, pp. 83 425–83 443, 2020.

[12] A. Subasi and E. Kremic, "Comparison of adaboost with multiboosting for phishing website detection," *Procedia Computer Science*, vol. 168, pp. 272–278, 2020.

[13] A. El Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *Ieee Access*, vol. 8, pp. 22 170–22 192, 2020.

[14] A. AlEroud and G. Karabatis, "Bypassing detection of url-based phishing attacks using generative adversarial deep neural networks," in *Proceedings of the sixth international workshop on security and privacy analytics*, 2020, pp. 53–60.

[15] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, 2019.

[16] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, vol. 11, no. 1, pp. 217–242, 2024.

[17] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn," *Electronics*, vol. 12, no. 1, p. 232, 2023.

[18] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid dnn–lstm model for detecting phishing urls," *Neural Computing and Applications*, vol. 35, no. 7, pp. 4957–4973, 2023.

[19] X. Xiao, W. Xiao, D. Zhang, B. Zhang, G. Hu, Q. Li, and S. Xia, "Phishing websites detection via cnn and multi-head self-attention on imbalanced datasets," *Computers & Security*, vol. 108, p. 102372, 2021.

[20] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw url and html characteristics," *Expert Systems with Applications*, vol. 236, p. 121183, 2024.