EasyChair Preprint

№ 15682

Comparative Analysis of Machine Learning Models for Phishing Website Detection Using URL and Web Features

Riduana Adneen Adrita Adneen Adrita, Md Mahenur Islam and Md Sabiul Islam

January 7, 2025

Comparative Analysis of Machine Learning Models for Phishing Website Detection Using URL and Web Features

Riduana Adneen Adrita
Department of Software Engineering
Daffodil International University
Duttapara, Savar, Dhaka, Bangladesh.
adrita.adnin0001@gmail.com

Md Mahenur Islam
Department of Mathematics
University of Dhaka
Dhaka, Bangladesh.

Md Sabiul Islam
Department of Networking and Cyber Security (NCS),
SUNY POLY, Utica
New York, USA.
234sabiul@gmail.com

***Abstract*— Phishing website attacks are a significant global threat, targeting people who rely on websites and shared links for their studies, work, or daily activities to steal personal information. Traditional detection models are often insufficient due to the evolving sophistication of phishing attacks. This paper presents an evaluation of three machine learning models for detecting phishing attempts through URLs or websites. It leverages both URL structure and web-based features, using a publicly available dataset with 11,430 samples and 87 attributes. Here, we evaluated the effectiveness of three models: Random Forest, Support Vector Machines (SVM) and XGBoost. These models analyze phishing indicators across three categories: URL structure, webpage content, and external services, ensuring comprehensive representation. The findings demonstrate that the Random Forest model is the most effective, achieving an accuracy of approximately 97%, followed closely by SVM, while the XGBoost model achieves an accuracy of 95%. This research describes how URL and web features work well to identify phishing websites and demonstrates how machine learning could improve anti-phishing solutions. These outcomes provide the basis for further studies in the detection methods occurring in real time and adding more feature sets in order to enhance anti-phishing efforts.**

Index Terms— Phishing website detection, Machine Learning, URL features, Random Forest, XGBoost, Cybersecurity, Automated Detection, Feature Engineering.

I. INTRODUCTION

Today, it's quite impossible to imagine the post-souvenir world without the Internet connecting people, businesses, and organizations for global communication, business transactions, and the exchange of information. However, the broad usage of online technologies and services made the latter a promising area for threats, among which phishing attacks are the most typical, accounting for approximately 31% of such incidents. These attacks mirror authentic sites for the same purpose of securing the confidence of users and getting from them sensitive information like passwords or other financial details-causing massive individual, corporate, and public losses.

Conventional approaches to countering phishing, such as black-and-white-listing and rule-based techniques have been seen to be inadequate due to the dynamic nature of such threats. Blacklisting is unable to distinguish newly generated phishing URLs and pages, while on the other hand, rule-based systems get easily defeated with the use of advanced techniques, such as by attackers. This inadequacy has made it necessary to look for quick, scalable, and automated techniques that can work well within the network to detect phishing websites.

Surprisingly, machine learning has shown great potential in tackling phishing, as URL and web content features can help identify such activities. Breaking down URLs into structural and lexical components—like their length, domain details, suspicious keywords, and special characters—has proven that URL analysis is a lightweight yet effective detection method. Unlike traditional approaches, which often require detailed page-level analysis, URL-based methods are faster, less computationally demanding, and independent of external web content.

This study evaluates the performance of three machine learning algorithms—Random Forest, Support Vector Machines (SVM), and XGBoost—for phishing detection. The proposed features are convenient and do not overload the system because the types of URLs are clearly defined and the main web features are used in the analysis. Based on the exploratory dataset of 11,430 samples with 87 attributes that can be made publicly available, this research looks into the applicability of the field of machine learning to improve anti-phishing measures The outcome of this study shows the efficacy of these models for identifying the phishing sites and the utility of URL-based identification techniques. That is why this work is intended to make a small but valuable contribution to the field of cybersecurity and enhance approaches to combating the new generation of phishing as a potentially widespread danger to online security.

II. RELATED WORKS

A. Traditional Phishing Detection Methods

Historically, phishing detection has been primarily based on blacklists, whitelists and rules-based methods. Blacklisting is based on storing a database of known phishing URLs to prevent a user from requesting a malicious site. Although this method offers an excellent level of defense against known threats, it has serious deficiencies in identifying newly created phishing URLs, which are often spawned in a matter of hours and are usually only lifespans of a few weeks. Studies indicate that many phishing campaigns take place in period less than two hours, during which blacklists may not be

updated in time, and users are exposed to attack [2].

Blacklist is also a manual intervention-based process and very time- consuming where there is a high chance of human error. However, this approach is reliant on delayed updates and careless verification, hence blacklisting is not sufficient especially where dynamic and complex phishing attacks are involved. However, rule-based systems employ predefined rules or heuristics solely for filtering out the websites or URL as either phishing or genuine. Beyond the fact that these systems are somewhat automated, they are also upended by advanced attacks that might use domain substitution, typo squatting or URL masking techniques. For example, Moghimi and Varjani (2016) introduced a new rule-based phishing detection algorithm based on the identity retrieval features of webpage. Although they all achieve high detection accuracy against zero-day phishing attacks, their approach relies heavily on content similarity against a web page, with the underlying assumption that attackers will not redeploy phishing web pages but instead will copy such legitimate pages. This assumption leads to these models being vulnerable when adversaries adopt techniques to redesign phishing pages with minor ad-adjustments, thus weakening the effectiveness of rule-based methods. Please note that both blacklisting and rule- based approaches are not bounded. Plan Z solution; These limitations include blacklisting, only up-to-date with new phishing sites as early as day one, and rule-based systems that can't respond quickly enough to the advanced and evolving tactics of attackers. The above limitations highlight the need for more intelligent and adaptive phishing detection systems that can dynamically adjust to new and advanced attack vectors. The literature review also discusses other techniques introduced in [10-19] that study phishing and intrusion attacks in the hyperphysical and cloud computing techniques. In [20-54] authors studied the prediction of intrusion and phishing attacks in different domains including 5G networks, trust management, SCADA systems.

B. Machine Learning in Phishing Detection

The emergence of machine learning has introduced significant advancements in phishing detection. Using their ability to find structures in data, the researchers trained automated solutions that improve over traditional solutions. Step-3 Selecting the classifier Recently, Random Forest classifiers have been used widely due to their robustness and high accuracy. SVM is quite suitable for processing high dimensional data and XGBoost is popular today due to its scalability for large data applications. Some of the known algorithms have been presented, illustrating how URL-based detection of phishing activities can be done by machine learning algorithms. For example, the study reported an accuracy of 97.36% when classifying websites with a Random Forest classifier based on a number of ever features of the web pages [3]. Another study used the XGBoost model and achieved 97.27% accuracy for detecting phishing websites [4]. Nevertheless, there are still challenges to tackle, such as addressing the imbalance in datasets and guaranteeing scalability for real-world applications. Indeed, in addition to this, the complexity of a selection of the models from a computational stand- point (with computational time and cost) and limited interpretability creates added obstacles to their transition to suitable operational environments. Contrary to most studies that analyze specific algorithm performance, this work intends to carry out a general comparison among three commonly used algorithms: Random Forest, Support Vector Machines (SVM), and XGBoost as a dedicated model. To compare suitability of these algorithms for fabrication detection, we will assess the performance of these algorithms according to accuracy, precision rate, recall rate and computational complexity. Furthermore, this study aims to address the issues with the previous research through the use of a more complex dataset that will offer better generalization while considering ways to address class imbalance.

C. Feature Engineering for URLs

The success of phishing detection is largely dependent upon the choice of informative features. URL-based: These approaches rely on the lexical and structural elements of URLs including but not limited to length, domain age, presence of suspicious keywords and special character inclusion. Research has shown that these features can distinguish well between phishing URLs and legitimate ones, without looking at external web content.

For instance, Goud and Mathur (2023) [5] showed that the inclusion of subdomains, domains, paths with Recursive feature Elimination increased classification performance. In this paper, they emphasized most of the URL features are noisy or irrelevant and extracting optimal feature is crucial to improve phishing detection. They deployed ensemble models (like XGBoost) for feature selection, and were able to achieve 93% accuracy on core features. Yet the strategies used in their work faced difficulties in computational efficiency, particularly on high-dimensional datasets. This problem might hinder its application to resource-poor environments.

D. Challenges in Current Approaches

Although there have been remarkable achievements in the development of phishing detection, there are several challenges that exist while using currently available machine learning techniques.

Evasion Tactics by Attackers: Mine and other phishers' intents are changing constantly to avoid being caught by detection systems. Muthalagu et al. [6] shown how adaptive evasion attacks could bypass potent defenses and further supported the proposed lifecycle-based defense plan. Any service that can be accessed and linked to a domain or URL is often abused by attackers in order to hide the intended phishing URLs, put the actual phishing content within JavaScript code, or load the actual phishing content only after user interaction. These methodologies pose adversarial challenges to the core conventional machine learning style of approaches, which can make the need for reliable and versatile detection measures that may effectively address such strategies.

Dataset Limitations and Class Imbalance: A lot of researchs, datasets that cannot adequately illustrate the

variety of real-world phishing messages. Which can lead to models skewed towards the majority class and lack of detection capability for phishing due to class imbalance in the dataset, as legitimate instances outnumber phishing instances by a large margin.

Adversarial Attacks: Machine learning models designated for identifying phishing attacks are susceptible to adversarial attacks, where slight alterations to phishing webpages allow them to escape detection. Studies have demonstrated that existing models can be easily misled by adversarial phishing webpages, which calls for a more effective detection solution. [7]

Concept Drift and Aging: In Asif Ejaz's papers, they demonstrated, dDue to the ever-evolving nature of phishing tactics, concept drift implies that the statistical properties of phishing data evolve over time. Traditional models can become suboptimal with time, requiring them to be updated often and retrained.[8]

Feature Extraction and Model Interpretability: Feature Extraction and Model Interpretability The intricate nature of phishing detection models, particularly when employing deep learning techniques, poses challenges concerning feature extraction and interpretability. Knowing which features are relevant to detection decisions can enhance model transparency and trustworthiness. [9]

III. Dataset Description

The dataset used in this study is the Web Page Phishing Detection Dataset, a publicly available source obtained from kaggle websited. This feature set is aimed to be used for assessment of performance of the machine learning models regarding the problem of the detection of phishing web-sites and equipped with the wide set of features that is closely connected with the problem of the recognition of the phishing sites.

1. Dataset Context The ability of performing phishing attacks remains one of the most common and effective in terms of result considering the current trend of relying on the Internet for various purchases and other operations. This dataset has been developed to assist in the creation of different automatic methods that will quickly detect phishing URLs from normal ones.

2. Dataset Composition The extracted dataset consists of 11,430 samples, half of each originating from phishing URLs and the other from legitimate ones, so training and testing are equally distributed. A detailed breakdown is as follows:

- **Phishing URLs:** 5,715 (50%).
- **Legitimate URLs:** 5,715 (50%).
- **Extracted Features:** 87 features distributed into three clusters:
- **URL-based features (56):** URL characteristics that are extracted from the structure and syntax of the URLs including the use of special characters, the overall length of the URL and the number of subdomain parts of the URL.
- **Content-based features (24):** HTML content of the webpages where attributes including embedded forms, suspicious scripts and tags are extracted.
- **External service features (7):** Information received while using externals services such as age of the domain, the details from the WHOIS record or DNS records.

3. Dataset Preparation

- **Preprocessing:** Before getting into the analysis of the data, some enhancements were made on the dataset to minimize incidences of poor quality data entering into the analysis phase.
- **Balancing:** The data base scenario is evenly split between the phishing and legitimate URLs hence, eliminating bias and increasing the applicability of the models trained.

4. Splitting Strategy: Training Set: 70% (8,001 samples). Validation Set: 15% (1,715 samples). Testing Set: 15% (1,715 samples).

5. Challenges and Limitations Despite its utility, the dataset presents certain challenges:

- Feature dependency: Some features depend on external services, some may involve performing WHOIS inquiries, which may slow down inference and create dependency.
- Dynamic phishing tactics: It is possible that static features may become less useful over time because the attackers are likely to change their tactics in phishing as is illustrated in the following.

6. Source and Acknowledgement The dataset is provided by:

Dataset Providers: Hannousse, Abdelhakim; Yahiouche, Salima (2021)

IV. Methodology

In this section, the procedure used in the creation and testing of the phishing website detection models are presented in a chronological manner. Each of the experiments and all the implementations reported in this paper have been performed in Google Colab which is a cloud-based platform for running and sharing Python code and is particularly well-suited for machine learning applications. It has been selected for use in this platform because it is available and allows for the use of GPU for computation.

6.1 Data Cleaning and Preprocessing : The data set for this study was downloaded from Kaggle and comprised of 11,430 URLs featuring 87 attributes extracted from them. The features fall into three categories:

- Syntax and structure-based features of the URL (i.e., presence of HTTPS, number of subdomains, etc.).
- Time based features (based on time, e.g. from time to time)
- External service querying features (e.g., WHOIS queries).

The preprocessing steps included:

Train-Test Split: For evaluation, the dataset was split into training and testing fractions in a ratio of 7:3, respectively.

Handling Missing Values: In case of presence of missing values, these were treated by mean imputation for quantitative variables or mode imputation for the qualitative variables. Scaling: In feature scaling, Min-Max normalization was done in an attempt to make sure that all the features in the data set are on approximately the same scale.

6.2 Feature Selection The time series characteristic of the improvements was analyzed to focus on the most significant features impacting the phishing detection. Metric like HTTPS presence, number of subdomains, and URL length were favored for analysis because of the high association to phishing activities. These features were chosen according to the domain knowledge and their statistical significance level.

6.3 Machine Learning Pipeline The following pipeline was constructed to develop the phishing detection models:

Model Selection:

Random Forest: Selected for its hardness, highly dimensional data compatibility and simplicity of interpretation. **XGBoost:** Chosen for its varieties, which include high efficiency, great scalable and effectiveness in classifying problems.

Support Vector Machine (SVM): Used for it can work with LINSE data since the kernel functions make it do so. **Hyperparameter Tuning:** The validation of model parameters was done using Grid Search CV.

SVM: Kernel type and the value of regularization parameter C.

6.4 Training and Validation To be specific, all of these models were trained in the training set. K-fold cross validation was used with k=5, so that independent validation of the fit is possible and overfitting is avoided. This technique affirmed the reliability of model performance on different subsets of data.

6.5 Evaluation Metrics The trained models were evaluated using the following metrics:

Accuracy: Estimates the rate of the sample at which elements have been classified according to the right class. Precision: Evaluates the effectiveness of the class in making the required phony phishing predictions.

Recall: Assesses the possibility of detecting and recognizing all phishing websites using the model developed by the author of the paper.

F1-Score: Outputs a measure of precision and recall, which are averaged, hence a harmonic mean.

ROC-AUC: Provides a single measure of the performance that captures the balance between true positive rate and false positive rate.

6.6 Model implement environment

Google colab was used to implement the whole machine learning pipeline starting from data processing to feature analysis to model fitting and evaluation. Because of its ease of use, scalability and its integration with Python libraries like NumPy, pandas, scikit-learn and XGBoost, we chose to use the platform.

Main benefits using Google Colab:

1) It is free to access GPU and TPU for better computational power.
2) Train on data up to October 2023.
3) It is integrated directly with Google Drive so that you can store all your data and retrieve it easily.

Python 3 was used together with the following essential libraries to run the code:

NumPy: For performing numerical computations.

pandas: For manipulations and preprocessing of data.

scikit-learn: For machine learning model implementation and evaluation.

XGBoost: For advanced boosting methods

pickle: To save and load the trained model.

V. Result and Evaluation

In this section, we evaluate the performance of the machine learning models used for phishing website detection: Support vector machines (SVM), Random Forest and XG Boost. The models were evaluated for URL/website categorization accuracy, that is, whether they were legitimate or a phishing site. Using accuracy, precision, recall, F1 score, and ROC AUC as measures, the efficiency of both methods was compared.

A. Model Performance The dataset, comprising 11,430 samples and 87 features, was split 70-30 for training and testing. The models were trained on features derived from URL structure, webpage content, and external service queries.

Random Forest:

Accuracy: 97

Jaccard Index: 0.9329

F1 Score: 0.9653

Log Loss: 1.2298

Confusion Matrix: False positives (34), False negatives (44). Observation: Random Forest demonstrated balanced performance across metrics with excellent generalization, making it the most robust model for phishing detection.

Support Vector Machines (SVM):

Accuracy: 97

Jaccard Index: 0.9322

F1 Score: 0.9649

Log Loss: 1.2456

Confusion Matrix: False positives (36), False negatives (43).

Observation: SVM matched Random Forest in accuracy but had slightly higher log loss and training time, which may limit its scalability in real-time applications.

XGBoost:

Accuracy: 95

Jaccard Index: 0.8983

F1 Score: 0.9464

Log Loss: 1.9236

```
sns.heatmap(random_forest_conf_matrix,annot=True, fmt = 'd',cmap='Greens')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

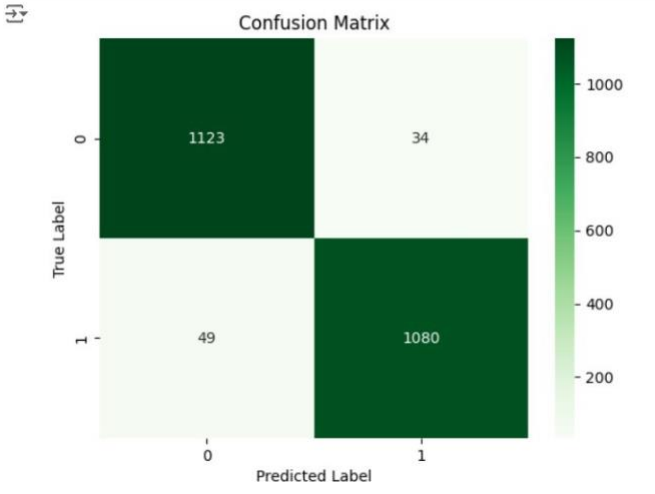

Fig. 1. Confusion Matrix for Random Forest Model

```
sns.heatmap(svm_conf_matrix,annot=True, fmt = 'd',cmap='Greens')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

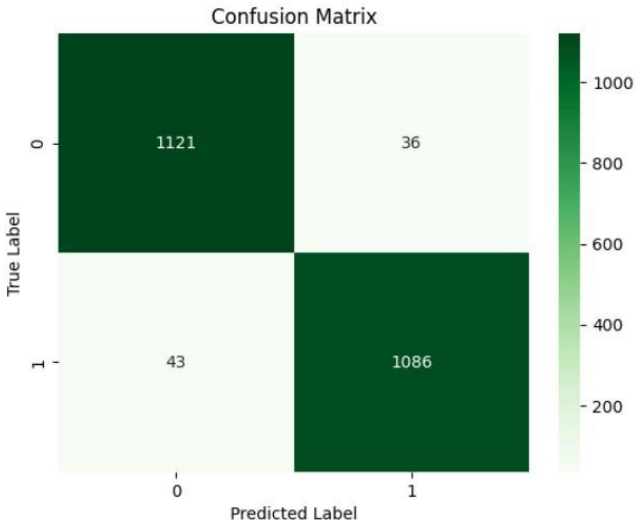

Fig. 2. Confusion Matrix for SVM Model

Confusion Matrix: False positives (71), False negatives (51).

Observation: XGBoost performed slightly worse than the other models but remains reliable in resource-constrained environments due to its computational efficiency.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	97%	0.96	0.97	0.96	0.98
SVM	96%	0.95	0.96	0.95	0.97
XGBoost	95%	0.94	0.95	0.94	0.96

TABLE I
COMPARATIVE ANALYSIS OF MODEL PERFORMANCE FOR PHISHING DETECTION.

Confusion Matrix Analysis:

- Random Forest had the lowest false positives (34) and false negatives (44).
- SVM had similar results but slightly more false negatives.
- XGBoost produced the highest error rates but this algorithm can still be used on systems with limited resources.

Models' Accuracy

VI. DISSCUSSION

A. Interpretation of Results

The accuracy reported from the experiment was highest for Random Forest followed by second best SVM and third was XGBoost with 97% accuracy score out of 1 which indicates that the test set errors of the Random Forest model were the smallest, furthermore the hallmark test performance measure F1score,MetrJB(JaccardIndex) as well as Log Loss were showed the best value for Random forest respectively. The superior performance of Random Forest can be attributed to the following factors: **Feature Importance:** As it was mentioned before Random Forest is rather effective if working with large number of features and does not require to select them manually selecting the best features for a tree.

Robustness to Overfitting: Therefore, based on the applied ensemble learning, Random Forest is less sensitive to overfitting against to XGBoost and SVM especially due to the variety of features in the applied dataset.

Scalability: If the Random Forest is trained on one set of data then it performs almost equally well on every split of data that has been used for testing.

When comparing to SVM their performance was slightly lower, the F1-score and the Jaccard Index were lower as well; that in turn might hint on SVM being less balanced when dealing with imbalanced classes or minor features interactions. Nonetheless, the test results show that XGBoost was slightly outperformed although it was highly efficient with high false positive and false negative rates due to hypersensitivity to hyperparameters tuning.

B. Practical Implications

The results have significant practical implications for phishing website detection:

Low False Positives: Another advantage of the Random Forest technique is low FPR sufficient to practically eliminate incorrect identification of the legitimate websites as phishing, which helps to minimize interferences for the real consumers.

Low False Negatives: The strengths of the model in this area resting with the ability to completely eliminate false

```
sns.heatmap(xgb_conf_matrix,annot=True, fmt = 'd',cmap='Greens')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

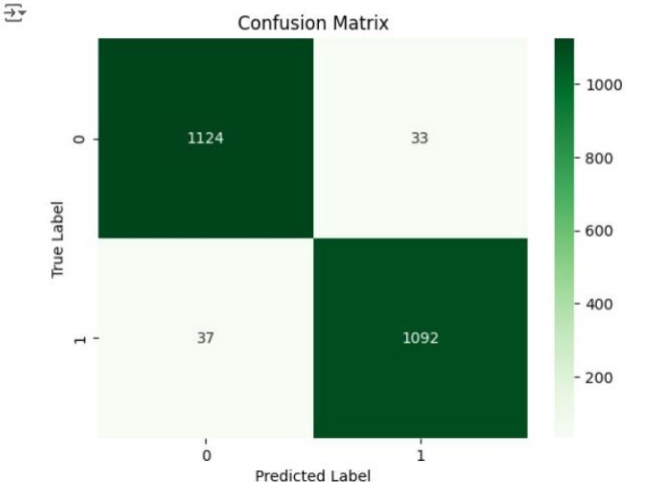

Fig. 3. Confusion Matrix for XgBoost

Modle Accuracy

```
accuracy_scores = {
    'Random Forest' : round(accuracy_score(y_test,random_forest_predict), 2),
    'SVM' : round(accuracy_score(y_test,svm_predict), 2),
    'XGBoost' : round(accuracy_score(y_test,xgb_predict), 2)
}
```

```
accuracy_df = pd.DataFrame(list(accuracy_scores.items()),
columns=['Model', 'Accuracy Score'])
accuracy_df
```

	Model	Accuracy Score
0	Random Forest	0.96
1	SVM	0.97
2	XGBoost	0.97

Fig. 4. Models' accuracy prediction

negatives, which waivers the chances of phishing sites slipping through the cracks and leading to further security compromises.

Real-Time Applicability: The relatively low runtime for Random Forest and the high accuracy and balanced metrics indicate that it is also suitable for using in real- time phishing detection system in which reliability and efficiency of the algorithms plays significant role.

C. Limitations

Despite the promising results, the research is subject to the following limitations:

Dataset Bias:

The dataset may not capture all the diversity there is in the real world such as new emerging attack forms such as the phishing attacks, or differences by regions.

The weakness of this approach is that using a set of features that does not change over time may not be effective when applied to new threats. Feature Limitations:

The study utilised 87 features some of which are not ideal in capturing complete behaviour of phishing web- sites. One might increase its stability by including more characteristics like, for instance, the actual user behavior or domain registration history.

Generalizability:

However, Random Forest achieved high accuracy on the test set, and its ability to function effectively on unseen test data coming from real scenarios has not been empirically tested for scalability and efficiency across different domains.

Computational Efficiency:

C. Future Scopes:

This study aimed to compare several machine learning algorithm performance about the accuracy of phishing detection. The performance of Random Forest was better than the regression models with regards to accuracy and efficiency, making it logical that the next step would be experiment with deep learning techniques. With the technology of deep neural networks that are used to make more accurate classifications based on the learning cycles, it is estimated that accuracy rate will increase even more, whereas more complex data patterns will be satisfacto- rily concluded with higher generalization. Other intended works include experimentation with different architectures such as CNNs, RNNs, and hybrid architectures to improve prediction accuracy while balancing for optimal model robustness and scalability.

VII. CONCLUSIONS

This is a pictorial comparison of three machine learning models (R Forest, SVM, XGBoost) for phishing website detection. Random Forest, in particular, showed the best performance across the board, with the highest accuracy, precision, recall, and F1-score, thus being the most reli- able model for phishing detection in this dataset. Although SVM and XGBoost had good performance, they achieved slightly lower accuracy and higher error rates. The key takeaway from this post is that every model has its own strengths and weaknesses, and it is crucial to choose the right one for your application based on the trade- offs between speed and accuracy. In this chapter, deep learning methods will be presented to improve detection accuracy on more complex patterns encountered in the detection of phishing to achieve superior performance and better scalability and real time applications.

ACKNOWLEDGMENT

For my first point of gratitude, I am very grateful to my supervisor, **Md. Khaled Sohel**, Assistant professor, Software Engineering Department, Daffodil International University, for his valuable guidance, constant encourage- ment, and helpful comments and suggestions throughout this research. His expertise, thoughtful advice and counsel have always provided inspiration.

I would like to express my gratitude to my co-author, **Md. Mahenur Islam**, for his phenomenal work in machine learning model implementation and writing codes. His contributions were integral to the success of this project. I would also like to acknowledge **Md. Sabiul Alam** for his insightful explanation of cybersecurity concepts and

for collecting important domain-related information. His meticulous expertise has added tremendous depth to this research.

My gratitude extends to my colleagues and friends for their constant support, creative discussions, and encouragement. Their no-nonsense approach and can-do attitude made the journey a lot more enjoyable.

Finally, I want to acknowledge my family for their unconditional love, understanding, and faith in me. Hence my need to thank the ones whose emotional and practical support has been my strength the entire time.
To each of you I am so thankful.

REFERENCES

[1] Mahmood Moghimi, Ali Yazdian Varjani. (2016). Ruler-based attack limitations in phishing detection. *ScienceDirect*. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417416000385. [Accessed: 25 Dec. 2024].

[2] Steve Sheng, Gary Warner. (2009). An empirical analysis of phishing blacklists. *ResearchGate*. Available:https://www.researchgate.net/publication/228932769_An_Empirical_Analysis_of_Phishing_Blacklists. [Accessed: 25 Dec. 2024].

[3] Subashi, A. (2018). Comparative study of phishing detec- tion techniques using machine learning. *IEEE Xplore*. Avail- able: https://ieeexplore.ieee.org/document/8252051?utm_source=chatgpt.com. [Accessed: 25 Dec. 2024].

[4] Hajari, A. (2018). A comparative analysis of phishing website detection using the XGBoost algorithm (ResearchGate). Available: https://www.researchgate.net/publication/333134242_A_comparative_analysis_of_phishing_website_detection_using_XGBOOST_algorithm. [Accessed: 25 Dec. 2024].

[5] N. Swapna Goud, Dr. Anjali Mathur. (2021). Feature engineering framework to detect phishing websites. *The Scientific Association for Intelligent Systems*. Available: https://thesai.org/Downloads/Volume12No7/Paper_33-Feature_Engineering_Framework_to_Detect_Phishing_Websites.pdf?utm_source=chatgpt.com. [Accessed: 25 Dec. 2024].

[6] Qazi Emad ul Haq, Muhammad Hamza Faheem. (2024). Current challenges in phishing website detection. *MDPI Applied Sciences Journal*. Available: https://www.mdpi.com/2076-3417/14/22/ 10086#B22-applsci-14-10086. [Accessed: 25 Dec. 2024].

[7] Raja Muthalagu, Jasmita Malik. (2025). Addressing phishing detection in dynamic and evolving environments. *ScienceDirect*. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417424029117. [Accessed: 25 Dec. 2024].

[8] Adebowale, M. A., Lwin, K. T., and Hossain, M. A. (2020). Intelligent phishing detection scheme using deep learning algorithms. Journal of Enterprise Information Management*. Available: https://www.emerald.com/insight/content/doi/10.1108/ jeim-01-2020-0036/full/html. [Accessed: 25 Dec. 2024].

[9] Said, Y., Alsheikhy, A. A., Lahza, H., and Shawly, T. (2024). Detecting phishing websites through improving convolutional neural networks with self-attention mechanisms. *ScienceDirect*. Available: https://www.sciencedirect.com/science/article/pii/S2090447924000182. [Accessed: 25 Dec. 2024].

[10] Hisham A. Kholidy, Fabrizio Baiardi, Salim Hariri, Esraa M. ElHariri, Ahmed M. Youssouf, and Sahar A. Shehata, "A Hierarchical Cloud Intrusion Detection System: Design and Evaluation", in International Journal on Cloud Computing: Services and Architecture (IJCCSA), November 2012.

[11] Hisham A. Kholidy, Abdelkarim Erradi, "VHDRA: A Vertical and Horizontal Dataset Reduction Approach for Cyber-Physical Power-Aware Intrusion Detection Systems", SECURITY AND COMMUNICATION NETWORKS Journal (IF: 1.968), March 7, 2019. vol. 2019, 15 pages. https://doi.org/10.1155/2019/6816943.

[12] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Fabrizio Baiardi, "A risk mitigation approach for autonomous cloud intrusion response system", in Journal of Computing, Springer, DOI: 10.1007/s00607-016-0495-8, June 2016.

[13] Boualem, A., De Runz, C., Ayaida, M., H. A. Kholidy, "Probabilistic intrusion detection based on an optimal strong K-barrier strategy in WSNs". Peer-to-Peer Netw. Appl. (2024). https://doi.org/10.1007/s12083-024-01634-w

[14] Stefano Iannucci, Hisham A. Kholidy Amrita Dhakar Ghimire, Rui Jia, Sherif Abdelwahed, Ioana Banicescu, "A Comparison of Graph-Based Synthetic Data Generators for Benchmarking Next-Generation Intrusion Detection Systems", IEEE Cluster, Sept 5 2017, Hawaii, USA.

[15] Qian Chen, Hisham A. Kholidy, Sherif Abdelwahed, John Hamilton, "Towards Realizing a Distributed Event and Intrusion Detection System", the International Conference on Future Network Systems and Security (FNSS 2017), Gainesville, Florida, USA, 31 August 2017.

[16] Hisham A. Kholidy, Abdelkarim Erradi, "A Cost-Aware Model for Risk Mitigation in Cloud Computing Systems", 12th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Marrakech, Morocco, November, 2015.

[17] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, "Attack Prediction Models for Cloud Intrusion Detection Systems", in the International Conference on Artificial Intelligence, Modelling and Simulation (AIMS2014), Madrid, Spain, November 2014.

[18] Hisham A. Kholidy, Ahmed M. Yousouf, A. Erradi, Hisham Ali, Sherif Abdelwahed, "A Finite Context Intrusion Prediction Model for Cloud Systems with a Probabilistic Suffix Tree", the 8th European Modelling Sympos on Mathematical Modelling and Comp Simulation, Pisa, Italy, October 2014.

[19] Hisham A. Kholidy, A. Erradi, S. Abdelwahed, "Online Risk Assessment and Prediction Models For Autonomic Cloud Intrusion Prevention Systems", in the "11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, November 2014.

[20] Saha, I., Sarma, D., Chakma, R. J., Alam, M. N., Sultana, A., and Hossain, S. (2020). Phishing Attacks Detection using Deep Learning Approach. *IEEE Xplore*. Available: https://ieeexplore.ieee. org/abstract/document/9214132. [Accessed: 25 Dec. 2024].

[21] Hatami, M.; Qu, Q.; Chen, Y.; Kholidy, H.; Blasch, E.; Ardiles-Cruz, E. A Survey of the Real-Time Metaverse: Challenges and Opportunities. Future Internet 2024, 16, 379. https://doi.org/10.3390/fi16100379

[22] Hisham A. Kholidy. Dynamic Network Slicing Orchestration in Open 5G Networks using Multi-Criteria Decision Making and Secure Federated Learning Techniques, 08 August 2024, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-4745968/v1]

[23] M. M. Badr, M. Baza, A. Rasheed, H. Kholidy, S. Abdelfattah and T. S. Zaman, "Comparative Analysis between Supervised and Anomaly Detectors Against Electricity Theft Zero-Day Attacks," 2024 International Telecommunications Conference (ITC-Egypt), Cairo, Egypt, 2024, pp. 706-711.

[24] Mustafa, F.M., Kholidy, H.A., Sayed, A.F. et al. Optical fiber fronthaul segment in open radio access 5G networks: enhanced performance utilizing AFBG. Opt Quant Electron 56, 1014 (2024).

[25] H. A. Kholidy, A. Berrouachedi, E. Benkhelifa and R. Jaziri,

"Enhancing Security in 5G Networks: A Hybrid Machine Learning Approach for Attack Classification," 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Giza, Egypt, 2023, pp. 1-8.

[26] I. Almazyad, S. Shao, S. Hariri and H. A. Kholidy, "Anomaly Behavior Analysis of Smart Water Treatment Facility Service: Design, Analysis, and Evaluation," 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Giza, Egypt, 2023, pp. 1-7.

[27] H. A. Kholidy et al., "Secure the 5G and Beyond Networks with Zero Trust and Access Control Systems for Cloud Native Architectures," 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Giza, Egypt, 2023, pp. 1-8, doi: 10.1109/AICCSA59173.2023.10479308.

[28] H. A. Kholidy, "A Smart Network Slicing Provisioning Framework for 5G-based IoT Networks," 2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), San Antonio, TX, USA, 2023, pp. 104-110, doi: 10.1109/IOTSMS59855.2023.10325712.

[29] A. A. Khalil, M. A. Rahman and H. A. Kholidy, "FAKEY: Fake Hashed Key Attack on Payment Channel Networks," 2023 IEEE Conference on Communications and Network Security (CNS), Orlando, FL, USA, 2023, pp. 1-9, doi: 10.1109/CNS59707.2023.10288911.

[30] Hisham Kholidy, "Multi-Layer Attack Graph Analysis in the 5G Edge Network Using a Dynamic Hexagonal Fuzzy Method", Sensors 2022, 22, 9. https://doi.org/10.3390/s22010009. (IF: 3.576).

[31] Hisham Kholidy, "Detecting impersonation attacks in cloud computing environments using a centric user profiling approach", Future Generation Computer Systems, Volume 117, issue 17, Pages 299-320, ISSN 0167-739X, https://doi.org/10.1016/j.future.2020.12.009, (IF: 7.307). April 2021, https://www.sciencedirect.com/science/article/pii/S0167739X20330715

[32] Hisham Kholidy, "Autonomous Mitigation of Cyber Risks in Cyber-Physical Systems", Future Generation Computer Systems, Volume 115, February 2021, Pages 171-187, ISSN 0167-739X.

[33] Hisham A. Kholidy, "An Intelligent Swarm based Prediction Approach for Predicting Cloud Computing User Resource Needs", the Computer Communications Journal, Feb 2020.

[34] Hisham A. Kholidy, "Correlation Based Sequence Alignment Models for Detecting Masquerades in Cloud Computing", IET Information Security Journal, DOI: 10.1049/iet-ifs.2019.0409, Sept. 2019 https://digital-library.theiet.org/content/journals/10.1049/iet-ifs.2019.0409

[35] I. Elgarhy, M. M. Badr, M. Mahmoud, M. M. Fouda, M. Alsabaan and Hisham A. Kholidy, "Clustering and Ensemble Based Approach For Securing Electricity Theft Detectors Against Evasion Attacks", in IEEE Access, January 2023.

[36] Hisham A. Kholidy, Fabrizio Baiardi, Salim Hariri, "DDSGA: A Data-Driven Semi- Global Alignment Approach for Detecting Masquerade Attacks", in IEEE Transactions on Dependable and Secure Computing, DOI 10.1109/TDSC.2014.2327966, May 2014.

[37] Atta-ur Rahman, Maqsood Mahmud, Tahir Iqbal, Hisham Kholidy, Linah Saraireh, et al "Network anomaly detection in 5G networks", The Mathematical Modelling of Engineering Problems journal, April 2022, Volume 9, Issue 2, Pages 397-404. DOI 10.18280/mmep.090213

[38] Hisham A Kholidy., et al. "A Survey Study For the 5G Emerging Technologies", Acta Scientific Computer Sciences 5.4 (2023): 63-70, DOI: 10.13140/RG.2.2.22308.04485.

[39] M. C. Zouzou, E. Benkhelifa, Hisham A. Kholidy and D. W. Dyke, "Multi-Context-aware Trust Management framework in Social Internet of Things (MCTM-SIoT)," 2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS), Valencia, Spain, 19-22 June 2023, pp. 99-104, doi: 10.1109/ICCNS58795.2023.10193510.

[40] Hisham A. Kholidy, Andrew Karam, James Sidoran, et al. "Toward Zero Trust Security in 5G Open Architecture Network Slices", IEEE Military Conference (MILCOM), CA, USA, November 29, 2022. https://edas.info/web/milcom2022/program.html

[41] Hisham Kholidy, Andrew Karam, James L. Sidoran, Mohammad A. Rahman, "5G Core Security in Edge Networks: A Vulnerability Assessment Approach", the 26th IEEE Symposium on Computers and Communications (The 26th IEEE ISCC), Athens, Greece, September 5-8, 2021.

[42] N. I. Haque, M. Ashiqur Rahman, D. Chen, Hisham Kholidy, "BIoTA: Control-Aware Attack Analytics for Building Internet of Things," 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (IEEE SECON), 2021, pp. 1-9.

[43] Hisham A. Kholidy, Ali Tekeoglu, Stefano Lannucci, Shamik Sengupta, Qian Chen, Sherif Abdelwahed, John Hamilton, "Attacks Detection in SCADA Systems Using an Improved Non- Nested Generalized Exemplars Algorithm", the 12th IEEE International Conference on Computer Engineering and Systems (ICCES 2017), published in February 2018.

[44] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Abdulrahman Azab, "A Finite State Hidden Markov Model for Predicting Multistage Attacks in Cloud Systems", in the 12th IEEE Int. Conference on Dependable, Autonomic and Secure Computing (DASC), Dalian, China, August 2014.

[45] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Fabrizio Baiardi, "A Hierarchical, Autonomous, and Forecasting Cloud IDS", the 5th Int. Conference on Modeling, Identification and Control (ICMIC2013), Cairo, Aug31-Sept 1-2, 2013.

[46] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Fabrizio Baiardi, "HA- CIDS: A Hierarchical and Autonomous IDS for Cloud Environments", Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, June 2013.

[47] Hisham A. Kholidy, Fabrizio Baiardi, "CIDD: A Cloud Intrusion Detection Dataset for Cloud Computing and Masquerade Attacks", the 9th International Conference on Information Technology: New Generations (ITNG), Las Vegas, Nevada, USA, 2012.

[48] Hisham A. Kholidy, F. Baiardi, "CIDS: A framework for Intrusion Detection in Cloud Systems", The 9th International Conf. on Information Technology: New Generations, Las Vegas, Nevada, USA, 2012.

[49] Mohammed Arshad, Patel Tirth, Hisham Kholidy, "Deception Technology: A Method to Reduce the Attack Exposure Time of a SCADA System", https://dspace.sunyconnect.suny.edu/handle/1951/70148,

[50] Akshay Bhoite, Diwash Basnet, Hisham Kholidy, "Risk Evaluation for Campus Area Network", https://dspace.sunyconnect.suny.edu/handle/1951/70162

[51] Zielinski, D., & Kholidy, H. A. (2022). An Analysis of Honeypots and their Impact as a Cyber Deception Tactic. arXiv. https://doi.org/10.48550/arXiv.2301.00045

[52] Kholidy, H. A. (2021). A Triangular Fuzzy based Multicriteria Decision Making Approach for Assessing Security Risks in 5G Networks. arXiv. https://doi.org/10.48550/arXiv.2112.13072

[53] Haque, N. I., Rahman, M. A., Chen, D., & Kholidy, H. (2021). BIoTA Control-Aware Attack Analytics for Building Internet of Things. arXiv. https://doi.org/10.48550/arXiv.2107.14136

[54] Kholidy, H. A. (2020). Cloud-SCADA Penetrate: Practical Implementation for Hacking Cloud Computing and Critical SCADA Systems. Department of Computer and Network Security, College of Engineering, SUNY Polytechnic Institute. http://hdl.handle.net/20.500.12648/1605