International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning-Based Phishing Detection System

Ahd Al-qasmi[1], Aseel Al-anazi[2], Lujain AL-shehri[3], Shoug A-lshaman[4], Wiam Al-atawi[5], Dr. Onytra Abbass*[6]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: Phishing attacks are still one of the most important threats to cybersecurity, exploiting human weaknesses to illicitly obtain sensitive information such as credit card numbers, personal data, and passwords. These attacks are generally carried out by misleading emails or websites that mimic legitimate sources, which can have severe consequences, such as financial losses, identity theft and data breaches within the organization. To address this growing concern, we have developed a phishing detection system using a random forest (RF) model. The model has been trained on significant Mendeley datasets_2020 and has demonstrated considerable advantages in accurately detecting phishing attempts. By analyzing the critical features of the site's URL, the system can distinguish between legitimate and malicious sites. Our comprehensive evaluation showed a high 99.4% accuracy and makes it a reliable tool for phishing detection. We have integrated the system into Chrome's web extension, allowing real-time detection and improving user protection. The paper highlights the potential of machine learning in cybersecurity and offers opportunities for future research and development to improve phishing detection through advanced ML techniques and larger, more diverse datasets.

Keywords: *Feature Extraction, Machine Learning, Mendeley Dataset 2020, Phishing Detection, Random Forest*

1. Introduction

Phishing is an online crime wherein a user is tricked into divulging personal information about themselves, potentially leading to identity theft. Hackers often use social engineering and technical techniques to pretend to be reputable organizations in phishing attacks. a particular kind of attack in which false websites are used to obtain unauthorized access to private data. The world's financial system is at risk from successful phishing attempts, which highlights the necessity of cybersecurity to fend off such attacks.

Phishing attack detection is the process of spotting phishing attacks early on, alerting administrators and users, and, ideally, reducing the threat. Phishing detection is always changing because attackers are always coming up with new strategies. Machine learning is one of the most effective ways to identify these malicious activities, but it also requires regular updates and maintenance, which is necessary for phishing detection measures to be effective. This is due to the fact that machine learning techniques can recognize certain common characteristics shared by the majority of phishing attacks.

By using machine learning's capacity to recognize patterns and abnormalities in data, it can assist in the detection of phishing attacks. Using it, models that automatically differentiate between reputable and nefarious emails, websites, or other types of communication can be developed. When examining the structure and content of URLs, machine learning models are able to detect suspicious features like the presence of multiple subdomains, the use of special characters, or unusually long URLs. These features can assist the model in distinguishing between phishing and authentic websites.

phishing is the name given to the cybercrime that involves tricking people into visiting phony websites and getting them to enter personal information, addresses, social security numbers, usernames, passwords, and anything else that can be made to look real. These websites trick users into visiting the website and entering their important credentials because they have content that is similar to that of the genuine websites and have a Uniform Resource Locator (URL) that is closely associated with them.**[1]**

This study highlights the use of machine learning, particularly random forest models, in detecting phishing attacks through URL feature analysis. The system we propose aims to improve real-time phishing detection and mitigate risks to users by leveraging ML techniques

[1] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0004-5878-4669
Email: ahdalqasmi@gmail.com
[2] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0005-6958-8994
Email: aseel.alanazi111@gmail.com
[3] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0003-4748-5076
Email: Lujain.alshr@gmail.com
[4] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0003-5438-6324
Email: shoogk--sa@hotmail.com
[5] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0009-8497-8068
Email: wiamalatawi@gmail.com
[6] *Department of Information Technology. University Of Tabuk –71411, KSA*
ORCID ID : 0009-0009-8497-8068
** Corresponding Author Email: obashir@ut.edu.sa*

1.1. .Background

Phishing has become a significant security risk in recent years, impacting both the targeted companies and people. This danger has been there for a while, yet it hasn't diminished in activity or effectiveness. Actually, in order to make their attacks more successful and believable, attackers have been refining their strategies throughout time. They target users with social engineering and exploiting the URLs in this attack. We will present a technique to detect this attack More specifically, we will use machine learning-based method, The model based on the random forest technique.**[1]**

2. Related Works

In **[2]**, Phishing Domain Detection Using Machine Learning: This study compared four models (ANN, SVM, DT, RF) using the UCI phishing dataset, applying MinMax normalization and five-fold cross-validation. Among these, RandomForests (RFs) are highly accurate at 97.76% the study highlights RF's reliability in phishing detection but points out the lack of temporal and dynamic features, emphasizing the need for more research on feature selection. Survey of Machine Learning-Based Phishing Detection Solutions.

In **[1]**, Deep Learning-Based Detection of Malicious URLs: This study delves into deep learning techniques for phishing detection. The PDRCNN model, which combines LSTM and CNN, achieved a 97% accuracy rate, providing effective, URL-based rapid detection but requiring significant training time and lacking consideration for website activity. The study also explored models like NISELM and hybrid methods with autoencoders, which improved feature extraction and dimensionality reduction. However, some models exhibited higher false positives and required more refined preprocessing.

In **[1]**, Machine Learning-Based Detection of Malicious URLs: Focusing on lexical URL features, this study achieved 99.7% accuracy using the Random Forest classifier on the set of data known as ISCX URL-2016. It showed the ability to independent phishing detection without relying on external services.

In **[3]**, Survey of Machine Learning-Based Phishing Detection Solutions: This study surveyed several phishing detection approaches. One approach using a Random Forest-based browser achieved 99.36% accuracy and integrated real-time phishing warnings, offering robustness and user-friendliness. However, it was tested on a single dataset (UCI). Additionally, CNN and PDRCNN models were compared, with PDRCNN attaining better accuracy (95.97%) and faster performance. While deep learning models extract intricate patterns from URLs, they are limited to URL strings, ignoring the dynamic nature of phishing websites. Hybrid models using Extra-Trees and meta-learners showed over 97% accuracy, though these models reduce data complexity, they were only tested on one dataset without comparison to advanced techniques.

In **[4]**, Phishing Detection Using Machine Learning Techniques: This research evaluated several machine learning algorithms (Logistic Regression, RF, SVM, XGBoost) on a dataset of 11,000 samples, concluding that XGBoost and RF delivered the best accuracy and overall performance. It provides a thorough analysis of phishing detection models but lacks discussion on the limitations of these models.

3. Methodology

In this Paper, we will go over the methodology that was employed as well as the system's structure and key components. These procedures consist of choosing the dataset, preprocessing it, splitting it, and then starting the model training process.

Table 1. Literature Review Evaluation

Model Used	*Dataset*	*Recall*	*Precision*	*Accuracy*
SVM	UCI phishing websites	96.6%	93.9%	94.66%
ANN		96.3%	95.6%	95.5%
RF		98.2%	96.9%	97.3%
DT		96.8%	96.5%	96.3%
RF	PhishTank website	98.14%	96.98%	97.26%
XGB		98.1%	98.72%	98.32%
VAE-DNN	ISCX-URL-2016	97.20%	97.89%	97.45%
RF	ISCX URL-2016.	-	-	98.7%
RF	UCI dataset	-	-	99.36%
extra tree& meta-learner				97%

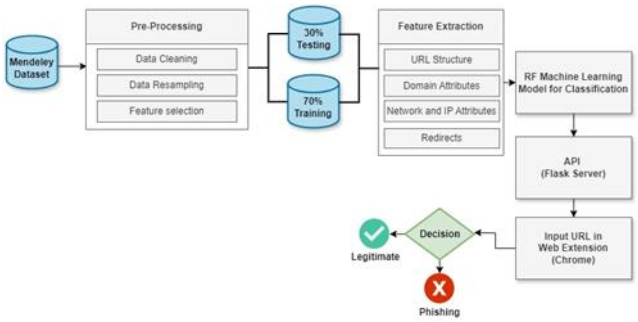

Fig. 1. The framework of the Proposed System.

As seen in **Fig 1**, the suggested system's framework was divided into multiple stages. URLs were first extracted from the Mendeley dataset (We chose the Mendeley_2020 dataset based on their numbers in instances and characteristics. Dataset Mendeley_2020 [5], it consists of two subdata sets: the full dataset and the small dataset. The full dataset contains 88,647 instances, while the small dataset has 58,645 instances. Data was gathered from PhishTank and Alexa rankings. The data set contains111 features, and we have classified them into eight groups. for better understanding, as described in **Table 2**.

Subsequently, the data underwent preprocessing, which included the following essential steps:

1. Data Cleaning: Identifying and correcting errors, handling missing values, and removing duplicates.
2. Resampling: Balancing the class distribution to prevent bias towards the majority class. [6]
3. Feature Selection: Reducing the number of features to simplify the model, shorten training time, and avoid overfitting.

This step also involved Dividing the data set into two groups:70% for train and 30% for test. After preprocessing, the next step was feature extraction, which focused on redirects, network and IP attributes, domain attributes, and URL structure.

For classification, the improved data was fed into a Random Forest (RF) machine learning model. Following this, the model was integrated with a Flask API to allow real-time URL input via a Chrome browser extension. Based on the classification results from the RF model, the system identified whether the URL was phishing or legitimate. If the URL was identified as phishing, the web extension redirected users to a secure HTML page.

Table 2. Dataset Features.[7]

Group	No.	Description	Type
1	1-17	each number of"-_/?=@&!~,+*#"$%" signs in the whole URL	Numeric
2	18-34	each number of"-_/?=@&!~,+*#"$%" in domain	Numeric
3	35-51	each number of"-_/?=@&!~,+*#"$%" in directory	Numeric
4	52-68	each number of"-_/?=@&!~,+*#"$%" in file	Numeric
5	69-85	each number of"-_/?=@&!~,+*#"$%" in parameters	Numeric
6	86-96	number of vowels, number of parameters, time_response, asn_ip, time_domain_activation, time_domain_expiration, number of resolved Ips, number of resolved NS, number of MX servers, Time-To-Live, number of redirects	Numeric
7	97-102	Top-level domain character length, number of characters in the whole URL, number of domain characters, number of directory characters, number of file characters, number of parameters characters	Numeric
8	103-111	is email present, is URL domain in IP address format, is "server" or "client" in domain, is TLD present in parameters, is domain has SPF, is URL has valid TLD/SSL certificate, is URL indexed on Google, is domain indexed on Google, is URL shortened	Boolean

3.1. Train Classifier Model

As in **Fig. 2.** Random Forest is an ensemble learning method that combines multiple decision trees to make predictions, in this model, each decision tree was trained on the entire Mendeley_2020 dataset, During the training process these individual trees learned to classify URLs based on various features, as shown in Figure 2. The strength of Random Forest lies in its ability to aggregate the predictions of these trees to make a final prediction, where the class that received the most votes was chosen. This method improved overall classification accuracy while reducing the effect of individual errors. Random Forest minimizes overfitting and enhances accuracy by using random feature selection. This approach reduces correlation between trees, promoting diversity and independence in predictions. By leveraging the combined strength of diverse decision trees, Random Forest achieves robust and accurate results.

Steps of Random Forest for Data Classification:

1. Use the entire Mendeley_2020 dataset for training each decision tree.
2. Create a Randomized Decision Tree by selecting a random subset of features at each split.
3. Aggregate the predictions of all decision trees, and the final output is the class with the majority vote.

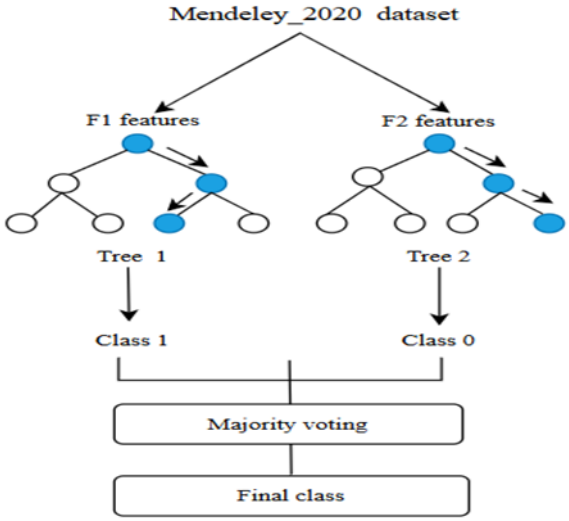

Fig. 2. Random Forest Model

3.2. Model Integration

- API Flask Server: Flask is a popular micro framework that is used to build web applications. We used Flask to create a REST API that allows you to get or send data to a website and perform some actions such as crud operations, using Flask offers benefits such as flexibility and simplicity, and using Flask to process HTTP requests coming from Chrome extensions and return appropriate responses. For example, when a user checks a URL, the extension sends a request to the Flask API, which processes it and returns the result. Flask also hosts server code for phishing detection logic. This includes loading pre-trained model, extracting features from URLs, and predicting whether URLs are phishing or legitimate.
- Web Extension: The integration of the phishing detection model into the web extension provides effective way to protect users from phishing attacks in real time and improve their security and online experience. We chose Chrome Browser because it is the most popular browser and works on different operating systems. Chrome is also known for its performance and speed.

3.3. Evaluation Metrics

To evaluate the effectiveness of the system in this Paper, we employed four distinct performance metrics: recall, accuracy, precision, and F1-score. Accuracy is an important metric for assessing the overall accuracy of classification models because it represents the correct predictor of the case. to all URLs in the dataset. A high accuracy shows that a mode can accurately predict both positive (Phishing) and negative (Legitimate) [8].

Precision gauges how well a model predicts positive outcomes. It measures the proportion of correctly predicted positive URLs to all correctly predicted positive URLs (false positives as well as true positives). Since precision evaluates the model's capacity to refrain from mistakenly labelling legitimate websites as phishing, it is especially important. A low false positive rate is implied by a high precision value.[8]

Recall Sometimes referred to as sensitivity or real positive rate, the model evaluates the ability to accurately identify each positive URL in the dataset. It determines the correlation between the correct positive prediction and all positive URLs. (including false negatives). Having a high recall rate is essential because it shows how well the model captures most real phishing URLs and reduces false negatives. The F1-score offers a fair assessment of a model's performance since it is the harmonic mean of precision and recall. It deals with scenarios in which both high recall and high precision are required. The F1 score is especially important in this context, where It is important to find a middle ground Between the accurate detection of phishing sites and the reduction of false positives. [8]

The last performance metric, classification time, is just the amount of time (measured in seconds) that a classifier needs after training on a particular dataset in order to predict the category of a new URL.

These metrics has the ability to distinguish between the positive/negative (phishing/legitimate) classes. recall, The accuracy, precision and F1 score are determined by taking into account four factors. potential results. shown in **Fig. 3.** of a detection model's prediction (Confusion matrix).

Term	Definition
True Positive (TP)	When the model correctly predicts the positive class (Phishing).
True Negative (TN)	When the model correctly predicts the negative class (legitimate).
False Positive (FP)	When the model incorrectly predicts the positive class (Phishing).
False Negative (FN)	When the model incorrectly predicts the negative class (legitimate).

Fig. 3. Confusion Matrix.

4. Results And Discussion

In this study, we implemented a phishing detection system using Machine Learning Technique, which achieved a notable accuracy rate of 99.40% in classifying URLs as either phishing or legitimate. We extensively tested the model with a diverse dataset that included both phishing and legitimate samples. The consistent results obtained from these evaluations confirm the robustness and reliability of the system.

4.1. Methods and Tools

We used Python to implement the system due to its widespread adoption in scientific computing, data science, and machine learning. Python's combination of productivity, performance, and clean APIs makes it an ideal choice for developing effective machine learning models. Additionally, its extensive libraries and strong community

support enhance its applicability. Some of the Python libraries that we use in the system are: [9]

1) Scikit-learn: is a well-known Python library used for machine learning, offering various tools and algorithms for tasks like model selection, classification, regression, clustering, and dimensionality reduction.

2) Numpy: is a numeric library using Python used for scientific computing and working with arrays, it provides a high-performance multidimensional array object and tools for working with these arrays.

3) Pandas: This is a Python library created for handling and analyzing data. It is widely utilized for importing and organizing datasets. It offers high performance and user-friendly data structures along with tools for data analysis in Python. [10]

4) Requests: is a Python library that is utilized for. send HTTP requests. It makes using Python to handle answers and submit HTTP requests easier. It allows include headers, cookies, authentication data, and arguments in requests.

5) Whois: library in Python is used to query and retrieve information about domain names and IP addresses. It allows users to obtain details such as the domain owner, registration dates, expiration date, and other relevant information.

4.2. Model Development

To enhance user accessibility, we integrated the system into a Chrome extension. This extension enables users to input URLs directly for classification. When a user enters a URL into the extension, the system processes the URL through the model and displays the result, indicating whether the URL is classified as phishing or legitimate. This integration significantly improves user experience, making it easier for individuals to utilize the system effectively.

4.3. Model Evaluation

The effectiveness of phishing detection models is assessed using various indicators. and visualizations. The following diagrams illustrate the model's effectiveness:

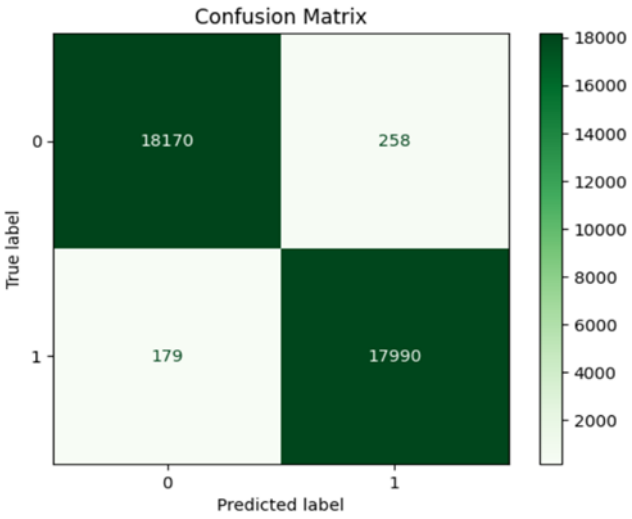

Fig. 4. Model's Confusion Matrix.

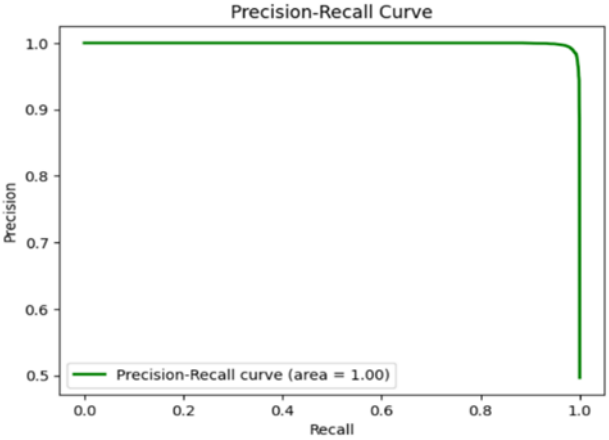

Fig. 5. Precision-Recall Curve.

As shown in **Fig. 5**, the AUC (Precision-Recall Curve) is 1.00, indicating that the model achieves perfect precision and recall across all thresholds. This exceptional performance suggests that the model is highly effective at distinguishing phishing URLs from legitimate ones, maintaining accuracy without significant false positives.

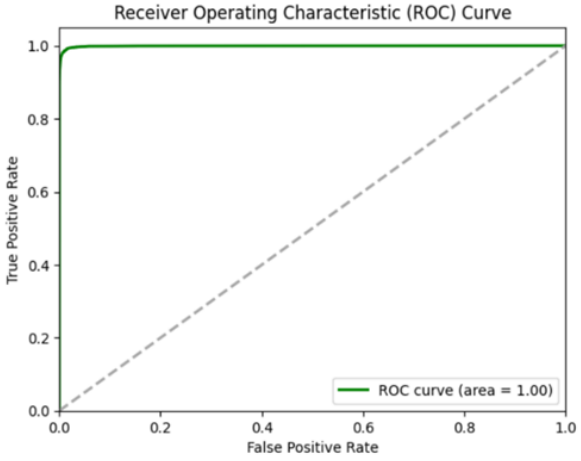

Fig. 6. ROC Curve.

The **Fig. 6**, shows the curve's proximity to the top-left corner, indicating near-perfect performance. This means the model is exceptionally effective at identifying phishing websites while minimizing the misclassification of legitimate sites as phishing.

From these values the metrics for evaluation the proposed model can be calculated as follows:

Precision: 98.59%

Recall: 99.01%

Accuracy: 99.40%

F1 Score: 98.80%

5. Conclusion

This paper introduces a machine learning-based phishing detection system, we reviewed different detection algorithms from both Machine-learning (ML) perspectives and presented the methodology and framework of model development. We chose the Random Forest algorithm, based on the evaluation results, and used the Mendeley_2020 dataset for implementation.

6. Future Works

In the future, we might also integrate "Continuous Learning in Machine Learning Systems" to let our system be more flexible and productive and continuously update itself without any retraining. We also plan an extension of the support on iOS and Android platforms.

Acknowledgements

We would like to thank everyone who contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Dr. ONYTRA ABBASS, for her invaluable advice, guidance, and her enormous patience throughout the development of the research. In addition, we would also like to express our gratitude to our loving parents and friends, who helped and encouraged us along the way.

Author contributions

Ahd Al-qasmi1, Aseel Al-anazi2, Lujain AL-shehri3, Shoug A-lshaman4, Wiam Al-atawi5: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Visualization. **Dr. Onytra Abbass6:** Supervision, Guidance, and Review.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, “An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders,” IET Inf. Secur., vol. 17, no. 3, pp. 423–440, 2023, doi: 10.1049/ise2.12106.

[2] S. Alnemari and M. Alshammari, “Detecting Phishing Domains Using Machine Learning,” Appl. Sci., vol. 13, no. 8, Art. no. 8, Jan. 2023, doi: 10.3390/app13084649.

[3] L. Tang and Q. H. Mahmoud, “A Survey of Machine Learning-Based Solutions for Phishing Website Detection,” Mach. Learn. Knowl. Extr., vol. 3, no. 3, Art. no. 3, Sep. 2021, doi: 10.3390/make3030034.

[4] V. Shahrivari, M. M. Darabi, and M. Izadi, “Phishing Detection Using Machine Learning Techniques,” Sep. 20, 2020, arXiv: arXiv:2009.11116. doi: 10.48550/arXiv.2009.11116.

[5] Y. Wei and Y. Sekiya, “Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection,” IEEE Access, vol. 10, pp. 124103–124113, 2022, doi: 10.1109/ACCESS.2022.3224781.

[6] H. Ali, M. Salleh, K. Hussain, A. Ullah, A. Ahmad, and R. Naseem, “A review on data preprocessing methods for class imbalance problem,” pp. 390–397, Oct. 2019, doi: 10.14419/ijet.v8i3.29508.

[7] G. Vrbančič, “Phishing Websites Dataset.” Mendeley Data, Sep. 24, 2020. doi: 10.17632/72ptz43s9v.1.

[8] S. Kapan and E. Sora Gunal, “Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features,” Appl. Sci., vol. 13, no. 24, Art. no. 24, Jan. 2023, doi: 10.3390/app132413269.

[9] S. Raschka, J. Patterson, and C. Nolet, “Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence,” Information, vol. 11, no. 4, Art. no. 4, Apr. 2020, doi: 10.3390/info11040193.

[10] V. Chang, V. R. Bhavani, A. Q. Xu, and M. Hossain, “An artificial intelligence model for heart disease detection using machine learning algorithms,” Healthc. Anal., vol. 2, p. 100016, Nov. 2022, doi: 10.1016/j.health.2022.100016.