JOURNAL OF MULTIDIMENSIONAL RESEARCH & REVIEW

http://www.jmrr.org Volume: 6, Issue: 2, April 2025 | ISSN: 2708-9452

Phishing URL Detection Using Machine Learning

Kamali S
MCA Student, PG & Research Department of Computer Science and Applications
Vivekanandha College of Arts and Sciences for Women [Autonomous],Tiruchengode
Namakkal, Tamilnadu, India.
Revathi J
Assistant Professor, PG & Research Department of Computer Sciences and Applications
Vivekanandha College of Arts and Sciences for Women [Autonomous],Tiruchengode,
Namakkal, Tamilnadu, India.

Abstract

As hackers utilise fake URLs to fool users into divulging personal information, phishing is a prevalent and hazardous issue in the realm of internet security. For this reason, the suggested method addresses phishing detection by utilising the Random Forest algorithm, a powerful ensemble learning technique. The Random Forest method combines the predictions of several decision trees to identify complex patterns in data and provide accurate classifications. To evaluate the legitimacy of URLs, a number of their characteristics are investigated using feature extraction. To evaluate the model's performance and ensure that it can effectively distinguish between phishing and legitimate URLs, key performance measures such as accuracy, precision, and recall are employed. By applying the benefits of the Random Forest algorithm to provide a robust solution to the persistent issue of phishing, the proposed study seeks to support the continuous efforts to enhance cybersecurity. Furthermore, the model's scalability and flexibility make it a viable approach for additional research and development in the field of URL categorisation. This technique's versatility across different datasets and scenarios emphasises how important it is for lowering phishing threats and improving overall security measures.

Keywords: Phising detection, URL classification, random forest, ensemble learning.

1 Introduction

Phishing is a prevalent and persistent issue in the realm of cybersecurity, when hackers pose as reliable websites to fool users into divulging personal information. This fraudulent activity often involves the creation of malicious URLs that imitate trustworthy online sources in an effort to trick users into disclosing personal information such as bank account details or login passwords. Traditional methods for identifying phishing attempts usually rely on manual verification, which is time-consuming and prone to errors. The rapid growth of internet usage and the steep increase in phishing tactics have made it imperative to develop automated systems that can recognize phishing URLs. Machine learning, in particular, and classification algorithms like Random Forest provide a potential approach to address this issue by analysing large datasets and identifying patterns that differentiate legitimate URLs from phishing ones. By using advanced algorithms, it is possible to create more reliable and efficient solutions for enhancing overall internet security and shielding users from phishing attempts.

1.1 Phishing Detection

Identifying fraudulent attempts to steal private information, such as credit card numbers, login passwords, or personal information, by disguising malicious websites as reliable ones is a critical step in the phishing detection process. This process include looking at several aspects of a website or URL, including domain names, URL structure, and behavior patterns. Phishing detection systems often employ machine learning algorithms that can learn from large datasets to distinguish between real and bogus URLs. Effective phishing detection may significantly reduce the risk of identity theft and financial loss by alerting users to potential threats before they connect with bogus websites.

1.2 Random Forest

The Random Forest method is an ensemble learning technique that creates a "forest" of decision trees, each of which is trained using a distinct subset of the data. The aggregate majority vote of the forest's trees serves as the basis for the computer's decisions. One of its key benefits is that it helps prevent overfitting by averaging out mistakes from several decision trees. Random Forest excels in handling complicated datasets and is capable of handling both classification and regression problems. Because it can recognize complex patterns in URL characteristics and categories them with high accuracy in phishing detection, Random Forest is a popular choice for security applications.

1.3 URL Classification

The technique of classifying URLs based on their characteristics to determine whether they belong to a legitimate website or a phishing site is known as URL classification. This process involves looking at things like the length of the URL, if any questionable keywords are included, the structure of the domain name, and whether HTTPS encryption is being used. In URL categorization, machine learning methods are commonly used to train models on large datasets, enabling them to recognise the traits that distinguish phishing URLs. To protect consumers from online threats, the goal is to accurately identify phishing attempts and automate the categorisation process. This can save the time and effort needed for manual URL verification while also improving overall security.

1.4 Ensemble Learning

Ensemble learning is a powerful machine learning approach that improves overall performance by combining the predictions of many models. The idea behind ensemble learning is that by combining many weak learners, a stronger, more accurate model may be created. Common ensemble learning strategies include bagging, boosting, and stacking, which differ in how they combine the results of the individual models. Random Forest uses bagging to train each decision tree on a different subset of the data, so implementing ensemble learning. The final projection is based on the sum of the output from each tree. Because ensemble learning boosts the model's resilience, reduces variance, and ensures better generalisation, it is especially useful in scenarios where high accuracy is crucial, such as recognising phishing URLs.

2 Literature Review

This research by Stefano Calzavara et al. surveys the most common attacks against online sessions, i.e., attacks that target genuine web browser users establishing an authenticated connection with a reliable web service. We then analyse existing security solutions that prevent or mitigate specific attacks using four different axes: protection, usability, compatibility, and simplicity of implementation. We also assess many defensive systems that are intended to provide robust defences against different types of assaults. We have determined five recommendations based on this survey, which the designers of the several concepts we examined have taken into account to varied degrees. We believe that these suggestions can help develop innovative solutions that address online security with greater methodicalness and thoroughness. The Web is the primary means of accessing online data and applications. It is extremely complex and diversified, including a large range of dynamic contents from several sources to deliver the optimum user experience [1]. According to Avinash Sudhodanan et al. Cross-Site Request Forgery (CSRF) attacks pose a serious threat to online systems. We focus on cross-site scripting (CSRF) assaults in this study, which target websites' identity management and authentication functionalities. All of these will be referred to as Authentication CSRF, or Auth-CSRF for short. We collected several Auth-CSRF attacks that have been reported in the literature, looked at their underlying strategies, and developed seven security testing methods that can help a human tester identify vulnerabilities that allow Auth-CSRF. To evaluate the effectiveness of our testing procedures and estimate the prevalence of Auth-CSRF, we conducted an experimental investigation on 300 websites from three different rank ranges of the Alexa worldwide top 1500. The results of our experiments are alarming: Of the 300 sites we examined, 133 were appropriate for our testing, and 90 of them (68%), had at least one vulnerability that made Auth-CSRF possible. Our testing methods were further developed, enhanced using the insights we gathered from our study, and included as an extension (CSRF-checker) into the open-source penetration testing tool OWASP ZAP [2]. Stefano Calzavara et al., implies that web sessions are fragile and susceptible to many types of attacks. Classic methods like as session hijacking, session fixation, and cross-site request forgery are particularly dangerous for online session security because they allow the attacker to undermine the integrity of legitimate users' sessions by forging requests that are authenticated on the victim's behalf. In this work, we analyse current defences against these assaults and their shortcomings, which might make them completely worthless based on specific assumptions about the attacker's abilities. We next extend our security study by offering black-box testing methodologies to identify vulnerable session implementation methods on existing websites using a browser plugin called Dredd. Finally, we use Dredd to assess the security of 20 popular Alexa websites, exposing many session integrity problems. Because the HTTP protocol is stateless, web applications by default handle each HTTP request independently of the others. However, online services often need to track state information over several HTTP requests to allow security-sensitive operations and to provide limited access to private information for authenticated users. In order to construct authorised web sessions, HTTP cookies are the most widely used method of preserving state information across HTTP requests [3]. Ayman Taha et al. claimed that the insurance sector is data-rich, with enormous volumes of customer data being analysed to determine risk. Machine learning methods are increasingly being used to efficiently control insurance risk. But by definition, insurance datasets are often low quality and contain noisy subsets of data (or features). Choosing the right data features is a crucial pre-processing step in the creation of machine learning models. The effectiveness of learning models has been demonstrated to be impacted by the addition of superfluous and duplicate features. In order to improve predictive machine learning techniques in the insurance sector, we propose in this study an approach for choosing relevant characteristics. Based on five real insurance datasets that are openly accessible, the experimental findings show how important it is to utilise feature selection to remove noisy characteristics prior to using machine learning algorithms. The algorithm will be able to focus on the most influential attributes as a result. An further economic benefit is the identification of the most and least important features in the datasets. These insights can be used to inform strategy and decision-making in areas and business concerns that are not directly related to the downstream algorithms. Our experiments demonstrated that machine learning techniques based on a subset of attributes suggested by feature selection algorithms outperformed the entire feature set for a collection of real insurance datasets [4]. Cross-Site Request Forgery (CSRF) vulnerabilities are a significant class of internet vulnerabilities that have received little attention from the security testing and research community (Johns, Martin et al.). The majority of CSRF vulnerability detection is still done by hand, despite the significant work that has been put into countermeasures like XSS and SQLi detection. In this study, we present Deemon, the first automated security testing framework to detect CSRF vulnerabilities to the best of our knowledge. Our approach relies on a new modelling paradigm that combines and fully graphs a number of web application components, including execution traces, data owes, and architecture layers. We provide the paradigm and show how a physical model with dynamic traces may be automatically constructed. Next, we use graph traversals to mine for potentially vulnerable operations. Our approach then automatically creates and executes security tests utilising the data stored in the model to successfully validate the identified CSRF issues. We evaluate Deemon's electiveness using 10 popular open source web applications. Our studies identified 14 hitherto unidentified CSRF vulnerabilities that may be used, for instance, to take over entire websites or user accounts [5].

2.1 Existing System

During a time when wireless communications are crucial for transmitting massive volumes of data, interference protection is crucial. Our study introduces a brand-new deep learning method for real-time phishing attack detection called the ResNeXt method with embedded Gated Recurrent Unit (GRU) model (RNT) [6]. Our methodical approach uses SMOTE to manage data imbalance during initial data processing, with a focus on strengthening digital forensics and combating the growing threat of phishing attacks. When autoencoders and ResNet (EARN) are combined with feature engineering, the model's discriminative performance is enhanced, especially during the feature extraction phase. Important patterns in the data are revealed using the ensemble feature extraction technique. The RNT model, which is the foundation of our AI classification, is hyper parameter-optimized using the Jaya optimization technique (RNT-J). Our AI model routinely outperforms state-of-the-art algorithms by a significant margin of 11% to 19% while maintaining remarkable computational efficiency, according to rigorous testing on real phishing attack datasets. In addition, our model has a mean execution time of 36.99s, a median of 35.99s, a minimum of 34.99s, a maximum of 41.99s, and a standard deviation of 1.10s. It also achieves 98% accuracy and low false positive/false negative values.

2.2 Proposed System

The Random Forest method, a potent ensemble learning technique that increases classification accuracy by mixing numerous decision trees, is used in the suggested system to improve phishing URL detection [7]. To differentiate phishing links from authentic ones, this system methodically examines a number of URL characteristics, such as length, domain structure, special character presence, and particular patterns. In order to guarantee a varied and organized collection of URLs for training, the procedure starts with dataset gathering. By addressing missing values, eliminating unnecessary information, and transforming categorical variables into numerical representations, preprocessing procedures are used to clean and format the data. In order to optimize the dataset for machine learning processing, feature extraction is essential for determining the most useful attributes that go into categorization. Following feature selection, the model goes through a training phase using labeled data, discovering correlations and patterns that aid in phishing attempt detection. Following training, the model's classification performance is assessed with unseen data to make sure it can generalize well across various URL types. Its efficacy is assessed using a number of evaluation criteria, such as accuracy, precision, recall, and F1-score, which reveal both its advantages and its shortcomings. By providing a scalable and flexible method for categorizing URLs according to phishing signs, the system seeks to contribute to the larger field of cybersecurity by delivering an automated and organized approach to phishing detection. This strategy is a useful tool for mitigating online dangers since it improves security measures by improving categorization techniques and decreasing reliance on manual intervention.

3 Module Description

3.1 Load Dataset

The system's "Load Data" module is the first stage where the dataset which is made up of different URLs is collected and ready for additional processing. It is essential in guaranteeing that a wide range of data is accessible for the system to leverage for learning. The machine can comprehend and distinguish between phishing and legal URLs thanks to this dataset. To prevent any biases in the model throughout the learning process, the module must make sure that the data is complete, balanced, and organized correctly.

This module may also entail importing data from various files or sources and transforming it into an appropriate format so that it can be analyzed.

3.2 Data Preprocessing

To prepare the dataset for the following stages, it must be cleaned and refined using the "Data Preprocessing" module. The performance of the model may be impacted by errors, missing values, or inconsistencies that are frequently present in the raw data [8]. By completing missing values, eliminating superfluous information, and transforming categorical variables into numerical formats that machine learning algorithms can readily use, this module addresses those problems. In order to guarantee consistency throughout the dataset, it could also entail scaling or normalizing specific data properties. By removing noise and inconsistencies from the data, this stage improves the model's capacity to learn and generate precise predictions.

3.3 Feature Extraction

The emphasis switches to locating and separating the most instructive characteristics from the URLs in the "Feature Extraction" module. The domain name, URL length, special character usage, and other patterns that might assist differentiate a genuine URL from a phishing one are just a few examples of the useful information that can be found in URL structures. In order to convert the raw data into a format that the machine learning algorithm can process efficiently, this module is made to extract these features from every URL. The system can simplify the dataset and concentrate on the elements that have the biggest impact on categorization by choosing the most pertinent features. To enhance the quality of the data entered into the model, it can entail adding new characteristics or integrating already ones [9].

3.4 Training and Testing

The system gains knowledge from the data in the "Training and Testing" module. A subset of the data is used to train a machine learning model, like Random Forest, after the features have been retrieved and the dataset is ready. The model gains the ability to recognize correlations and patterns between the characteristics and their respective URL labels. Following training, a distinct subset of the data that has never been seen before is used to test the model. By doing this, the model is guaranteed to be able to generalize effectively[10]. The testing stage makes it possible to assess the model's performance objectively and offers information on how well it can identify URLs. This stage is essential for estimating the model's likelihood of performing effectively in the presence of fresh, untested data.

3.5 Model Evaluation

The goal of the "Model Evaluation" module is to assess the model's performance following training and testing. This stage uses a number of metrics, including accuracy, precision, recall, and F1 score, to assess how well the model classifies phishing and authentic URLs. A thorough view of the model's advantages and disadvantages is given by these metrics. For instance, accuracy indicates the proportion of correct predictions made by the model, whereas precision and recall provide further information on the model's capacity to identify phishing URLs without producing false positives. The assessment procedure aids in pinpointing potential weak points in the model, directing future enhancements. Before the model is put into use, it also acts as a last validation to make sure it satisfies the necessary performance requirements.

4 Result Analysis

To find out how successfully the proposed model identifies URLs as genuine or phishing, its performance is evaluated using a variety of criteria. After training and testing, the model's outputs are compared to the actual labels in the test dataset to determine its accuracy and reliability. Important measures including accuracy, precision, recall, and F1 score are used to gauge the model's ability to correctly identify phishing URLs while lowering false positives and false negatives. While accuracy displays the total percentage of correct classifications, precision measures the proportion of predicted harmful phishing URLs. While recall focusses on how well the model detects phishing URLs, the F1 score balances precision and recall to offer a single measure of the model's overall performance. By looking at these results, we can assess the model's strengths and weaknesses in terms of phishing attempt detection, ensuring that it is precise and efficient. This study also helps guide future model enhancements, including altering features or improving the learning process, to increase classification accuracy. The accuracy of phishing URL identification has significantly improved when comparing the suggested and current methods. With an accuracy of 80%, the current system does reasonably well in categorization, although there is still opportunity for improvement. The suggested method, which uses the Random Forest algorithm, on the other hand, obtains 100% accuracy, proving its superior capacity to differentiate between phishing and authentic URLs. The model's capacity to evaluate several features at once, which lowers misclassification errors and boosts overall reliability, is responsible for this increase. The Random Forest algorithm's ensemble learning methodology improves pattern identification and decision-making, resulting in more accurate classifications. In comparison to the current method, the suggested technique efficiently reduces false positives and false negatives, guaranteeing a more accurate and efficient phishing detection mechanism, as evidenced by the significant accuracy increase.

5 Conclusion

In conclusion, the proposed method demonstrates how phishing URLs may be successfully identified using machine learning, particularly the Random Forest algorithm. The method uses a number of features of URLs to distinguish between malicious and legitimate connections, offering a reliable and efficient means of lowering phishing risks. The model's high level of accuracy in recognising phishing URLs is demonstrated by its performance as evaluated by metrics like accuracy, precision, and recall. Even while the current system is promising, it might be further developed and improved by including additional features or by refining the model to make it more resilient. Ultimately, this tactic offers a helpful tool in the ongoing efforts to enhance internet security and protect users from phishing frauds. In future research, the approach may be further enhanced by looking at alternative hybrid models or machine learning strategies that can boost its effectiveness and precision in detecting phishing URLs. Experimenting with a larger variety of features, such as complex text analysis techniques or domain-related patterns, might give the model more discriminative power. Moreover, expanding the dataset to include a wider variety of URLs from other sources might improve the model's generalisation ability and resilience. Using state-of-the-art techniques like ensemble methods or deep learning might help enhance performance, especially when managing phishing attempts that are getting more complex and sophisticated. The model may also be enhanced by utilising cross-validation techniques and modifying hyperparameters to ensure better performance on unknown data. These advancements might lead to a more comprehensive and reliable phishing detection system.

References

[1] M. Bugliesi, A. Ragazzo, A. Rabitti, and S. Calzavara, "Checking web sessions for integrity issues," in *Proc. 24th European Symposium on Research in Computer Security (ESORICS)*, Luxembourg, Sep. 23–27, 2019, pp. 606–624.

[2] G. Tolomei, A. Rabitti, R. Focardi, S. Calzavara, and M. Conti, "Mitch: A machine learning method for CSRF vulnerability blackbox detection," in *Proc. IEEE European Symposium on Security and Privacy (EuroS&P)*, Stockholm, Sweden, Jun. 17–19, 2019, pp. 528–543.

[3] M. Tempesta, M. Squarcina, R. Focardi, and S. Calzavara, "Web session security: A voyage to survive the web," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 13:1–13:34, 2017.

[4] U. Morelli, A. Armando, N. Dolgin, L. Compagna, R. Carbone, and A. Sudhodanan, "Extensive examination and identification of cross-site request forgeries in authentication," in *Proc. IEEE European Symposium on Security and Privacy (EuroS&P)*, Paris, France, Apr. 26–28, 2017, pp. 350–365.

[5] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 94:1–94:45, 2017.

[6] G. Pellegrino, M. Johns, S. Koch, M. Backes, and C. Rossow, "Deemon: Detecting CSRF with dynamic analysis and property graphs," in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, Dallas, TX, USA, Oct. 30–Nov. 3, 2017, pp. 1757–1771.

[7] A. Haines, M. Lalmas, F. Silvestri, and G. Tolomei, "Interpretable predictions of tree-based ensembles via actionable feature tweaking," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Halifax, NS, Canada, Aug. 13–17, 2017, pp. 465–474.

[8] K. Mahm, M. H. Khanmohammadi, S. Yazdani, and M. Mohammadi, "Financial reporting fraud detection: An examination of data mining procedures," *Int. J. Financ. Manag. Account.*, vol. 4, pp. 1–12, 2020.

[9] G. Stiglic, P. Kocbek, N. Fijacko, M. Zitnik, K. Verbert, and L. Cilar, "Interpretability of prediction models grounded in machine learning in the healthcare industry," *arXiv preprint* arXiv:2002.08596, 2020.

[10] D. Collaris and J. J. van Wijk, “ExplainExplore: Visual exploration of machine learning explanations,” in *Proc. IEEE Pacific Visualization Symposium (PacificVis)*, Tianjin, China, Jun. 3–5, 2020, pp. 26–35.

Cite this article:

Kamali S & Revathi J, “Phishing URL Detection Using Machine Learning”, Journal of Multidimensional Research and Review (JMRR), Vol.6, Iss.2, pp.219-227, 2025