Smart URL Threat Detection System Using Django and Supervised Learning

[1] Amith Kumar B, Student, MCA, P.E.S Institute of Technology and Management, Shivamogga, Karnataka, India.

[2] Dr. Sanjay K S, Associate Professor & Head, Dept. of MCA, P.E.S Institute of Technology and Management, Shivamogga, Karnataka, India

Abstract

Another common cybersecurity problem is the so-called phishing which frequently comes out in the form of misleading web links that are meant to steal valuable pieces of information about the user. In this project, we can suggest solving the problem of detecting such malicious URLs with the help of machine learning, based on Logistic Regression, Random Forest classifiers. The model analyses both the lexical pattern of the pages as well as its statistical characteristics of each URL to determine the legitimacy of it. Web application built using Django gives the user an opportunity to type in URL and get the immediate feedback in the form of classifications by the two models. The excellent levels of accuracy and reliability have been confirmed by performance analysis that indicates well-performing consistency of the system in real-life situations.

Keywords*: Malicious Link Detection, Supervised Learning, Random Forest, Logistic Regression, Django Framework, Online Security.*

1. Introduction

Phishing is the most common types of social engineering attack which often involves the use of misleading web links with the purpose of resembling genuine websites and, therefore, making the user give away his/her confidential information. With such attacks increasing both in frequency and complexity it is becoming apparent that the older traditional rule based detection systems are no longer adequate. Contrary to this, the machine learning techniques are characterized by an extra amount of flexibility and data-driven character, as they can be used to examine particular characteristics in what have been called phishing URLs. This paper proposes a two-model detection method, in the style of Logistic Regression and Random Forests classifiers, implemented as a smooth extension to a web interface based on Django to categorize phishing URLs in real-time. More old-fashioned solutions such as blacklists or manually created rule sets have a hard time detecting phishing links generated for the first time, or disguised in a clever manner. In addition, even though the content-based approaches, e.g. those suggested by Zhang et al. (2007) and Whittaker et al. (2010) may have accuracy, they can be too costly to implement on a real-time basis.

In order to cover these gaps, other researchers such as Mohammad et al. (2014) and Sahingoz et al. (2019) have shown a promising result in machine learning-based strategy revolving around lexical and statistical properties of the URLs. Riding on such innovations, this paper develops a system based on these models

in order I would like to a high possibility of discovering phishing. The novel technology which was developed does not only guarantee the high accuracy level and performance level of operation, but also focuses on the usability of the created technologies as its capabilities enable the end-user to engage in the system in a real-time mode, which makes academic solutions applicable in practical defense against cybersecurity attacks.

2. Literature Survey

[1] In Mohammad et al. (2014), this analysis the machine learning effectiveness in detecting phishing websites was considered by including the lexical and the host-related features of the URLs. They included models that examined feature importance; J48 or Naive Bayes and Support Vector Machines (SVM). They explained feature importance as a way of enhancing the accuracy of detection. Such features as availability of IP addresses and unusual lengths of the URL were discovered to play a vital role. The results should imply that the automated detection system, designed on carefully selected features, can be regarded as a reliable means of detecting phishing threats with almost no manual control.

[2] Zhang, Hong and Cranor (2007) also designed the approach of content-based method of detecting the phishing sites, named CANTINA (identifying the phishing sites by examining the text-based context of the web pages). It applied Term Frequency 1 Inverse Document Frequency (TF-IDF) to identify the significant word and this was compared to the real domain sources. This solution brought a significant change in past trend of black listing approach towards the manner of analysis of semantic content and it was this approach that ultimately proved to be good in terms of phishing sites detection.

[3] Sahingoz, et. al. (2019) conducted a thorough study on a machine learning based method to identify phishing with special attention to lexical attributes of URLs. They went through a careful process of selecting over 20 differentially interesting features from the URL, such as the length of the URL, occurrence of dots, use of special characters to train their classification models. They tried different algorithms against the data like Logistic Regression Compared to the Decision Tree, the ensemble-based forest model demonstrated improved precision and minimized false alarms. The study highlights the usefulness of other simple, URL-based features in phishing detection without relying on the content of the destination web site.

[4] Le et al. (2018) propose the technology URL Net which is a deep neural network architecture that would help deal with the issue of malicious URLs identification and learn to extract representations based on raw character- and word-level representations as their input. It does not involve handmade feature engineering because it can learn both small-scale and more general patterns of structure of URLs using convolutional neural networks (CNNs). The results of the experiments also indicated that the URLNet performed better compared to the regular machine learning based models, especially in case of complex or obfuscated

phishing URLs. A more emphasis was made in the analysis on the advantages of using an end-to-end deep learning models in order to improve detection of malicious links.

[5] Whittaker, Ryner and Nazif (2010) introduced a scaleable phishing detector framework that was developed by Google using the mixture of both URL based and content based attributes. The system is in continuous training and updating with the real life data hence the system is adaptable to the dynamic architecture of phishing attacks. The integration of the logistic regression and measures of blacklisting that were already there gave good results to the model in both accuracy and efficiency of the operations. This article clarifies that there is a need to maintain dynamic and frequently maintained detection models.

[6] Whittaker, Ryner and Nazif (2010) introduced a scaleable phishing detector framework that was developed by Google using the mixture of both URL based and content based attributes. The system is in continuous training and updating with the real life data hence the system is adaptable to the dynamic architecture of phishing attacks. The integration of the logistic regression and measures of blacklisting that were already there gave good results to the model in both accuracy and efficiency of the operations. This article clarifies that there is a need to maintain dynamic and frequently maintained detection models that will be in a position to recognize phishing menace in an effective and timely manner.

[7] Chiew, Yong, and Tan (2018) has done an extensive survey and analyzed the different types of phishing attacks, their method of delivery, and what to use to detect. The paper categorizes phishing into a number of categories including spear phishing, email phishing and clone phishing and evaluates the advantages and shortfalls of various defense mechanisms, including blacklisting, heuristic analysis and machine-learning. The survey is a good source of information to know how phishing mechanisms have evolved and will continue to be an essential tool to the study of cybersecurity.

[8] Feng et al. (2018) developed a customized neural network architecture aimed at identifying phishing websites by incorporating both structural and lexical attributes. Their approach utilized characteristics such as domain-related information, URL functionalities, and specific patterns in HTML tags as input features for model training. Compared to conventional classification techniques, the neural network delivered superior performance in terms of precision and recall. This study highlights the effectiveness of deep learning in detecting intricate phishing behaviors, ultimately contributing to more accurate and dependable threat identification.

[9] Rao, Ali, and Amaity (2015) studied whether a machine learning algorithm can detect phishing attempts based on lexical features contained in the URLs. In their work, they used these supervised algorithms Support Vector Machines, Decision Tree classifiers and in their case, they emphasized on speed of detection without loading and analyzing entire web pages. The models showed impressive scores, and this suggests that simple feature analysis on URLs would make a reliable and expeditious method of detecting phishing risks in real-time.

[10] Patil and Patil (2015) explored the Phishing URL detection was carried out using machine learning techniques that relied on 14 handcrafted features extracted from the URL's structure. Various classification models were evaluated, including Naïve Bayes, Support Vector Machines (SVM), and Random Forest. Among these, Random Forest demonstrated the highest accuracy and produced the least number of false positives. Their methodology emphasized preprocessing techniques such as normalization and feature selection. The research confirmed that machine learning techniques can efficiently identify phishing threats using only the characteristics of URLs.

[11] Jain and Gupta (2017) introduced a client-side security approach that leverages an auto-updating whitelist of verified domains to guard against phishing attacks. Their system is designed to instantly block unknown or suspicious URLs, prioritizing ease of use and performance over algorithmic complexity. Unlike machine learning models, this solution does not involve intensive training but instead offers a lightweight and efficient method for real-time phishing prevention. While it lacks adaptability, its simplicity and responsiveness make it a practical alternative for end-user protection.

3. Proposed Methodology

The goal of the proposed system is to accurately identify phishing URLs using machine learning approaches—specifically, Random Forest and Logistic Regression by analyzing only the structural and lexical features present in the URL.This approach eliminates the need to access or analyze webpage content, resulting in a fast, efficient, and resource-friendly detection process. Its lightweight nature makes it suitable for deployment in real-time, high-throughput cybersecurity environments.

The system's workflow begins by collecting a dataset comprising both phishing and legitimate URLs. Each URL is then processed to extract important lexical and structural features, including overall length, special characters (such as '@' or '-'), the presence of IP addresses, the count of dots, HTTPS protocol usage, and the age of the domain. These features are selected relying on past researches which confirm that they are effective in differentiating phishing links. Upon extraction, the data is treated to be encoded and normalized in order to be compatible with the machine learning models. Then the dataset will be split into two parts using training and testing parts. Independently, this prepared data is used to train both of the classifiers, Logistic Regression, and Random Forest. The Logistic Regression is used because it is computationally cheap and easy to interpret and easily draw conclusions as to how each predictive variable contributes towards the prediction. Conversely, Random Forest is used because of its attribute of dealing with a complex relationship of features and high accuracy. Then, to make the decision on performance of each of the given models easier, a number of evaluation metrics will be used, such as accuracy, precision, recall, F1-score, and ROC-AUC, to compare the performance of these models in the most adequate way.

To apply the trained models in a real-world context, a web-based platform is built using the Django framework. This user interface enables individuals to input URLs and instantly receive classification

results, including confidence levels from both machine learning models. The application not only validates the practical applicability of the system but also serves as a real-time, accessible solution that translates academic research into everyday cybersecurity protection.

3.1 Proposed model Diagram

The proposed system's architecture outlines the complete workflow of phishing URL detection through machine learning techniques. It takes users through each stage—from secure login and URL input to prediction and result presentation—via an intuitive and interactive web interface. The system conducts real-time evaluation by utilizing both Logistic Regression and Random Forest models, enabling prompt and precise identification of harmful URLs.

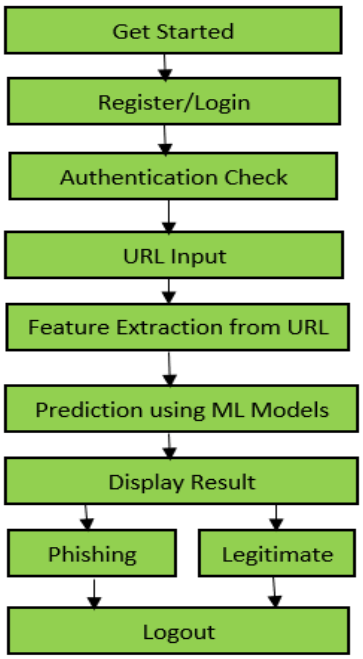

Fig: 3.1.1 Proposed model Diagram

The phishing detection process starts when a user logs into the system, ensuring that access is secure. After successful authentication, the user submits a URL for evaluation. The system will do analysis on a URL which will be submitted by pulling out key features thereof to include the total length of the URL, character specific characters, the IP address application and domain based details. The derived attributes are then analyzed against two machine learning classifiers Random Forest and Logistic Regression to come up with the fact that the URL is legitimate or might produce adverse effects. The result, plus additional confidence can be shown in an easy viewable way. After revising their outcome, the user logs out with the result of a secure and streamlined detection cycle.

5. Result

Random Forest and Logistic Regression are both machine learning techniques are commonly used in determine the functionality of phishing URL detection mechanism; experiments were conducted using the two supervised machine learning algorithms to judge the performance of proposed phishing URL detection system. As a measure of assessment, The key classification metrics—Accuracy, Precision, Recall, and F1-Score—were taken into account to offer a balanced evaluation of both models Such metrics are used To evaluate how effectively these models distinguish between phishing and legitimate URLs, emphasis was placed on their accuracy and predictive reliability in real-world conditions.

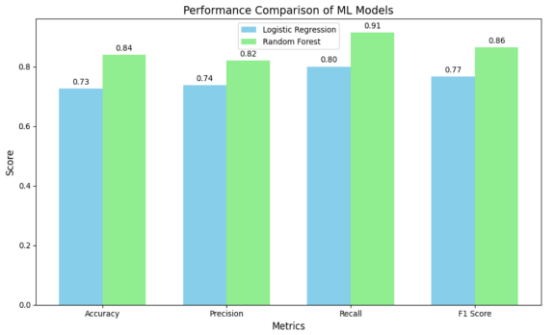

Fig 5.1.1 Display the Result of two model comparison

```
--- Logistic Regression ---
Accuracy : 0.7258
Precision: 0.7368
Recall   : 0.8
F1 Score : 0.7671

--- Random Forest ---
Accuracy : 0.8387
Precision: 0.8205
Recall   : 0.9143
F1 Score : 0.8649
```

Fig 5.1.2 Snapshot of Performance Matrix

Analysis

Logistic Regression achieved 72.58% accuracy, with a precision of 73.68% and recall of 80.00%, showing moderate effectiveness in detecting phishing URLs. However, Random Forest performed with 83.87% accuracy, 82.05% precision, and a higher recall of 91.43%. Its F1 Score of 86.49% indicates a strong balance, making it more reliable for phishing detection.

Model	Accuracy	Precision	Recall	F1Score
Logistic Regression	72.58%	73.68%	80.00%	76.71%
Random Forest	83.87%	82.05%	91.43%	86.49%

Conclusion

A comprehensive analysis of ten academic works on phishing URL detection highlights machine learning's essential contribution to cybersecurity enhancement. The study has reviewed a variety of supervised Classification techniques like tree-based models, logistic-based classifiers, ensemble forests, and support vector-based algorithms (SVM) were employed, and ensembles of different types. In the academic literature, the researchers were united in their attention to proper feature selection and data preprocessing procedures. Another notable conclusion of this literature review includes the findings that ensemble and tree-based methods tend to perform better as compared to linear analysis, a factor that has been accredited to the ability of the methods to handle complex and non-linear interactions among features.

Similar to these scholarly notes, our empirical findings reveal that the algorithm of Random Forest, Logistic Regression, and the rates of their correctness were 83.87 and 72.58 percent correspondingly. Furthermore In addition to achieving a higher Area Under the Curve (AUC), the model also showed enhanced performance across other key evaluation metrics, such as precision recall and the f 1 -score that have reaffirmed the results of the rest of the studies by extension supporting the robustness and versatility of Random Forest in proficiently categorizing phishing URLs.

Thus the system created in the work suggests the power of the ensemble-based methods in creating secure phishing detection methods eventually contributing to the final online security of the user.

Future Work

Although the existing system shows satisfactory performance using classifiers like Logistic Regression and Random Forest, there remains significant scope for improvement to further enhance its effectiveness. The next possible solution is applying deep learning algorithms (e.g. LSTM, CNN, or transformers), along with the ability to employ a kind of complex pattern learned directly on the string of the URLs to prevent handcrafting features. In addition, the external sources of information that can be imported to feed the system, such as domain reputation score or a threat intelligence API, can contribute to the effectiveness of the detection because such will present current information timely.

We can also develop the science by experimenting with hybrid models, or by adopting the stacking approach in which has several algorithms used to become stronger and precise. Further exploration of the phishing threats may be realized with the inclusion of such content-related attributes like JavaScript behaviour or HTML structure combined with lexical characteristics.

For practical deployment, integrating the system into browser extensions or offering it as a cloud API can support real-time detection with high user accessibility and minimal delay.

References

[1] Mohammad, R. M. F. & McCluskey, L. (2014). "An assessment of features related to phishing websites using an automated technique. "IEER

[2] Zhang, Y., Hong, J. 1., & Cramer, L. F. (2007). "CANTINA: A content-based approach to detecting phishing web site

[3] Salinger, O. K., Buber, E., Demir, O., & Diri, B. (2019). "Machine learning based phishing detection from URLS

[4] Le, H. Pham, Q. Sahoo, D., & Hoi, S. C. H. (2018). "URLINE Learning a URL representation with desp, learning for malicious URL detection

[5] Whittaker, C., Ryner, B., & Nazif, M. (2010). "Large-scale automatic classification of phishing pages. "NDSS.

[6] Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). "A survey of phishing attacks: Their types, vectors and technical approaches."

[7] Feng. F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018).
"The application of a novel neural network in the detection of phishing websites."

[8] Rao, R. S Ali, S. T., & Amaity, A. (2015). "A machine learning approach for phishing detection using URL features.

[9] Patil, D. R., & Patil, J. B. (2015). "Detection of phishing URLs using machine learning techniques. "ICCUBE

[10] Jain, A. K., & Gupta, B. B. (2017). "A novel approach to protect against phishing attacks at client side using auto-updated white-list."

[11] Abu-Nimeh, S., Nappa, D., Wang, X., comparison of machine learning techniques fo

[12] Basnet, R., Mukkamala, S., & Sung, A. H phishing attacks: A machine learning approach.

[13] Dunlop, M., Great, S., & Shelly, D. (201 images for content-based phishing analysis"

[14] Prakash, P., Kumar, M., Kompella, R. R. "PhishAst: Predictive blacklisting to detect phis

[15] Almomani, A., Gupta, B. B., Wan, T Manickam, S. (2013). "Phishing dynamic e framework for online detection zero-day phish

[16] Khonji, M., Iraqi, Y., & Jones, A. (2013). literature survey. "IEEE Communications Surve

[17] Afroz, S., & Greenstadt, R. (2011). "PhiskZ websites by looking at them."

[18] Starera, S., Provos, N., Chew, M., & Ru framework for detection and measurement of ph [19] Liu, W., Deng, X., Huang, G., & Fu @xppiising strategy based on visual similarity a [20] Zhang, H., Liu, G., Chow, T. W. & Liu, W visual content-based anti-phishing: A Bayesian

[21] Ereud., A., & Karabatis, G. (2012). "A co detecting malicious URLs using machine learni

[22] Ramesh, G., Krishnamurthi, I., & Kuma efficacious method for detecting phishing well domain identification."

[23] Marchal, S., Saari, K., Singh, N. & Asokan, phish: Novel techniques for detecting phishing.