© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714451

Phishing Detection Using Machine Learning: A Data-Driven Approach to Enhancing Cybersecurity Awareness

DIGVIJAY PURKAYASTHA
Department of Statistics and Data Science, Christ University, Bengaluru

Abstract—Phishing is still a significant and evolving cybersecurity threat for businesses as these types of attacks can often bypass rules and regulatory-based filters. Although there have been recent advances in deep learning techniques that provide a high accuracy in detection rates, there continues to be significant issues with the "black box" nature of these techniques resulting in a lack of interpretability, therefore making them difficult to deploy in a trustworthy manner within the cybersecurity environment. Therefore, this paper presents a new data-driven phishing detection framework that balances high detection accuracy and provides feature interpretability. Using the UCI Machine Learning Repository dataset, we evaluated multiple supervised learning algorithms such as: Logistic Regression, Support Vector Machines (SVM), and Random Forest and incorporated Synthetic Minority Over Sampling Technique (SMOTE) to correct for class imbalance. Our evaluations show that the Random Forest classifier provides the best performance with an accuracy of 95.5% and ROC-AUC score of 0.97, outperforming the other models that were evaluated. More importantly, we extend the analysis of phishing beyond a binary classification and report insight into the importance of features through the analysis of the Random Forest classifier. Specifically, our findings indicate that indicators such as Having_IP_Address and URL_Length are strongly predictive of phishing versus legitimate intent. Our research suggests that lightweight interpretable ensemble models can be developed that are scalable and transparent alternatives to large complex neural networks for real-time phishing detection.

I. INTRODUCTION

Phishing attacks are a significant cybersecurity challenge, utilizing techniques of social engineering to induce users to disclose sensitive personal information by creating an illusion of security through a presentation of legitimacy. Because of the high number of phishing messages sent, and the comparatively low expense for the attacker, phishing is a widely distributed, low-value form of cyber-attack. The ongoing history of major, widespread phishing attacks against both individual users and corporate organizations illustrates that an effective phishing detection capability is necessary to safeguard both personal and corporate assets against phishing attack vector activity.

A. Aim and Objectives

This study aims to develop and implement a machine learning model that will assist in identifying and preventing phishing attacks by analyzing structured public data from existing datasets. By creating a more efficient, scalable solution for protecting organizations against the threat of phishing, it will provide an empirical basis for developing and refining cybersecurity efforts.

Objectives of the project include:

•Analysis of phishing attack characteristics

•Research into how phishing attacks are created and grow

•Interpretation of a machine-learning model's outputs and the importance of inputs or features

•Development of new applications and future opportunities for using machine-learning models to fight against phishing.

B. Scope of Research

This study aims to detect phishing attacks by employing publicly available, structured datasets, which provide labelled examples of both legitimate and phishing websites. The use of Classical Supervised Machine Learning Algorithms (i.e., Random Forest, Support Vector Machine (SVM), and Logistic Regression) is aimed at identifying phishing attacks based upon interpretability, Computational Efficiency, and Table Data Structure, whereas advanced Deep Learning approaches were not addressed because of their high Computational Power requirement relative to the limitations of this Scope of Work.

II. LITERATURE REVIEW

In the most recent years, the prevalence of Phishing Attacks has received significant attention as a notable Cyber Threat, prompting researchers to focus their efforts on creating scalable and Efficient Phishing Detection Methodologies.

A. Traditional Approaches

Traditionally, heuristic rules or blacklists have been commonly used for Phishing Detection, however these methods are not without their limitations, requiring the constant updating of the manual list, and are unable to provide Zero-Day detection capability.

B. Machine Learning Approaches

Researchers have turned to new approaches to phishing detection by combining data with machine learning (ML) algorithms.

•Aburrous et al. (2010) developed a fuzzy logic and neural networks-based intelligent system to detect phishing sites using a set of rules; however, their system was limited in functionality and therefore impractical to implement.

•The model created by Mohammad et al. (2012) demonstrated that combining multiple input features into one feature set improved the model's performance significantly over previous models that did not use multiple input features.

•In their work on using Machine Learning classifiers to detect phishing sites, Patil and Patil (2015) were able to show that as well as having high detection rates, Decision Trees and Naïve Bayes classifiers also had very good levels of accuracy when used for phishing site classification.

•Jain and Gupta (2018) found that theRandom Forest algorithm provided higher importance weightings for the features used in their dataset as well as an overall accuracy score of over 90%. This study specifically pointed to the advantages of using ensemble learning methods to classify phishing URLs.

IV. GAPS IDENTIFIED

Although the literature identified key improvements have been achieved, there are still several gaps in how phishing is currently detected:

High False Positive Ratios: Many current methods produce accurate results through aggressive labeling, which causes legitimate websites to be labeled as phishers. This can adversely affect end users and lead to decreased confidence in the validity of a security solution.

Lack of Interpretability: Because deep learning algorithms are "black box" in operation, it is vital for cybersecurity professionals to understand why particular sites were flagged. For example, was it flagged because the site utilized an IP address or because the SSL certificate was deemed unsatisfactory?

Excessive Computational Overhead. Methods that monitor web pages using information from both the HTML source and images tend not to produce output quickly enough to provide real time scanning of sites within a web browser.

Failure to Correctly Address Issues of Imbalance: Most phishing data sets are extremely imbalanced, having significantly more legitimate than phishing sites, and this imbalance was not adequately addressed in most of the previous studies conducted. As a result, bias has been introduced into many models.

This work addresses all of these deficiencies through the utilization of a Random Forest Classifier which provides a combination of high accuracy, interpretability (through the use of feature importance), computational efficiency and, through the use of SMOTE, methods to mitigate the effects of data imbalance.

III. METHODOLOGY

This research employs a well-defined methodology encompassing the step-by-step creation of a phishing detection apparatus. The procedure includes the gathering of data, the processing of the information, the design of a model, the assessment of the model's performance, and the interpretation of the findings.

A. Data Collection

This research employs the UCI Machine Learning Repository's Phishing Websites data set as the source for its analysis. There are over 11,000 records in this data set, with each record representing an identified website that was classified "phishing" or "legitimate". Attributes assigned to these records relate to over 30 types of behavior exhibited by the URL, the Domain, and the Page from which the page touches the web.

B. Feature Description

Features are broken down into groups according to their characteristics:

1. Address Bar Features

• Having_IP_Address This feature examines whether the URL is using an IP address instead of a domain name. A URL that uses an IP address would be deemed suspicious (a value of -1).

• URL_Length This feature calculates the length of a URL. A URL that is excessively long would be viewed as suspicious (a value of -1).

• Shortining_Service This feature looks for URL shorteners (e.g., bit.ly39).

• Having_At_Symbol This feature identifies whether the URL contains an '@' symbol.

• Double_slash_redirecting This feature looks for occurrences of a double slash (//) in the URL.

2. Domain Features

• SSLfinal_State This feature evaluates whether or not the SSL certificate is valid and whether or not HTTPS is being used.

• Domain_registeration_length This feature verifies how long the domain is registered (less than one year would be considered short-term).

• Favicon This feature verifies if the favicon is being pulled from the same domain.

3. HTML and JavaScript Features

• Request_URL This feature counts the number of objects (images, scripts) that are being loaded from other domains.

• URL_of_Anchor This feature checks to see if the anchors link to an external/suspicious URL.

• Links_in_tags This feature counts the number of suspicious links found in meta tags, script tags and link tags.

• SFH (Server Form Handler) This feature indicates how the HTML form handler behaves.

C. Data Preprocessing

Here is a more accessible version of the entire text using language of lower complexity/more common usage.

1. URL Based Features

If URL uses IP address instead of domain name, it is suspicious.

If the length of the URL is too long, it is suspicious.

Using a URL shortening service (like bit.ly) is suspicious.

The presence of an '@' symbol in the URL is suspicious.

Having '//', or double slashes in the wrong place is suspicious.

2. Domain Based Features

The SSL certificate and use of HTTPS are evaluated for their validity.

A short term domain registration is less than one year.

The favicon is loaded from a different domain.

3. HTML and JavaScript Features

Counts the number of files that are loaded from other domains.

Checks to see if the anchors or links go to an external or suspicious URL.

Counts the number of links in 'meta', 'script', and 'link' tag types.

The HTML form handler has specific behaviors.

C. Data Preparation

Cleaning and preparing data for analysis consists of:

Identifying and/or replacing missing values

Converting categorical features (e.g., Yes/No) into numbers (e.g., -1/1)

Rescaling of features using scale conversion methods

D. Model Development

The algorithms available for model implementation include:

Logistic Regression (LR): A linear model that produces a baseline model that provides baseline performance information;

Support Vector Machines (SVM): A high-dimensional classification algorithm that utilizes the RBF kernel;

Random Forest (RF): A random forest is an ensemble model composed of a collection of decision trees; a random forest combined with multiple decision trees increases robustness and decreases variance.

Decision Trees (DT): A decision support tool used for determining basic decision rules; however, decision trees are also susceptible to overfitting.

The implementation of each of these algorithms is accomplished with Python and various libraries: NumPy, Pandas, and Scikit-learn. The model training and testing were performed on an 80-20 split with 5-fold cross-validation for model generalisability.

IV. RESULTS AND DISCUSSION

Using the UCI Phishing Websites Dataset, the machine learning models were assessed . Standard classification metrics were used: Accuracy, Precision, Recall, F1 Score, and ROC-AUC.

A. Model Performance Comparison

The results of this research study were based upon the results of evaluating four classification algorithms by using the UCI Phishing Website Dataset and using five-fold cross validation. The performance metrics evaluated were Accuracy, Precision, Recall, F1 Score, and ROC AUC. The Random Forest classifier was determined to be the best performing algorithm.

Table I: Performance Comparison of Classifiers on UCI Dataset.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	87.6%	85.1%	88.2%	86.6%	0.91
Decision Tree	91.3%	89.9%	91.7%	90.8%	0.93
SVM (RBF Kernel)	92.8%	91.5%	92.4%	91.9%	0.95
Random Forest	95.5%	95.1%	94.6%	95.0%	0.97

The Random Forest Model exhibited high precision and recall rates, with the model identifying 95.5% of phishing cases correctly from the dataset on which it was tested. As such, Random Forest is expected to provide very good precision and reliability in real-world situations involving phishing detection.

B. Feature Importance

Random Forest provided the capability to rank features based on their predictive ability; this is one of the primary advantages of the Random Forest model. The three features identified as having the greatest significance in predicting phishing attacks were Having_IP_Address, Request_URL, and URL_Length. These features support previous research into phishing methods, including obscuring legitimate domains and modifying form handlers.

C. Key Findings

SSL Clarification of SSL Misconceptions:

Phishing sites had legitimate SSL Certificates. Hence, it is a misconception that "HTTPS = Secure."

Domain Age:

The Domain_registration_length feature demonstrates that phishing sites have relatively short lifespans; that is, most phishing sites have relatively low Domain_Ages.

False Positives:

Legitimate sites were tagged as "phishing" because of their extensive use of third party scripts and redirections; these behaviours mimic the behaviour of phishing sites.

V. CONCLUSION

The research discussed here has demonstrated how supervised machine-learning algorithms can accurately identify phishing URLs using structured behavioral data. The analysis indicated that the Random Forest classifier provides a good balance between performance and computational efficiency, with the highest level of accuracy achieved being 95.5%.

Unlike "black-box" deep-learning methods, our model gives you valuable information about the structure of a hack when it is implemented with our feature importance methodology. Among other things, the primary indicators of phishing activity are technical anomalies identified in the feature importance analysis, including the use of an IP address in a URL, and the existence of links whose lengths are outside the norm. This interpretability is important when it comes to limiting false positives and establishing trust in automated security tools.

As a result, this study provides more evidence that standard machine-learning algorithms, when properly tuned and balanced, are capable of achieving the same accuracy as deep neural networks, but with much less computational cost. Future efforts will focus on integrating this lightweight detection tool into real-time browser extensions for rapid and clearly understandable detection of threats to end users.

Future Work

The work done in the present study forms a good basis for future research, with a number of specific areas which merit further investigation:

Implementation of real-time integration with web browsers: The system implemented in this research currently operates on pre-collected data, therefore, in future work a REST API

will be developed and the implementation added to a Chrome extension to allow for real-time blocking of phishing websites.

Development of deep learning-based approaches: Areas for further work that could be pursued are analysing the sequence of characters in URL addresses using Long Short-Term Memory networks (LSTMs), similar to how traditional NLP tasks are undertaken.

Incorporation of visual content analysis: By integrating CNNs into our solution to enable the comparison of visual images of websites against known brands, we can identify, for example, whether a website visually imitates a brand (for instance, PayPal), even when the URL does not match that of the brand.

REFERENCES

[1] UCIMLRepo Developers, "ucimlrepo Python package," Python Package Index (PyPI), 2023. Available: https://pypi.org/project/ucimlrepo/

[2] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL Detection using Machine Learning: A Survey," arXiv preprint, arXiv:1701.07179, 2017.

[3] Aburrous et al., "Intelligent phishing detection system for e-banking using fuzzy data mining," 2010.

[4] Mohammad et al., "An assessment of features related to phishing websites using an automated intelligent framework," 2012.

[5] Jain and Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," 2018.

[6] Scikit-learn documentation: RandomForestClassifier. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html