

PHISHNET: Multi-Algorithmic Safety Net for Advanced Phishing URL Detection

Mrs Anusha M¹, Prince Kumar², Sadhana Kumari³, Mohd. Arsalan Lari⁴, P.Sirisha⁵

¹Assistant Professor, Visvesvaraya Technological University, Computer Science, SJB Institute of Technology, Bangalore, Karnataka, India

^{2,3,4,5}Students, Visvesvaraya Technological University, Computer Science, SJB Institute of Technology, Bangalore, Karnataka, India

ABSTRACT

Today's generation heavily relies on web technology for various tasks such as banking and communication. However, this also means that users are exposed to numerous security threats, with phishing being one of the most serious and common attacks. Phishing attacks aim to steal sensitive information from users by impersonating a legitimate entity. Attackers use phishing to obtain victims' credentials such as bank account numbers, passwords, or other private information. The victim may not be aware of the website that is phishing, which impersonates a genuine website. Therefore, this paper proposes a system to identify phishing sites using machine learning algorithms. The system utilizes a classifier that is trained on a comprehensive dataset with enriched features.

Keywords: Phishing URL Detection, Machine Learning, Websites

I. INTRODUCTION

The Internet has grown to be an essential component of our lives, but it additionally makes it easier to do harmful acts like phishing discreetly. Phishers also try to deceive their victims by using the methods of social engineering or creating fake websites in order to get sensitive information from individuals and companies, including login details, account IDs, and passwords. Although various methods have been brought in to detect phishing websites, hackers continue to find ways to evade detection. Machine Learning is one of the most efficient techniques for identifying these detrimental actions. This is due to the fact that machine learning techniques can recognize some common traits shared by the majority of phishing assaults. In recent years, the advent of machine learning (ML) has revolutionized cybersecurity by offering automated and adaptive solutions for threat detection and mitigation. ML algorithms are highly suited for real-time phishing URL detection because they can evaluate huge amounts of information, identify patterns, and generate precise predictions. The present research offers a thorough analysis of the most recent machine-learning techniques for phishing URL recognition. We explore various features and algorithms employed in ML-based approaches, ranging from simple lexical analysis to sophisticated models.

II. Literature survey

a) A novel algorithm to detect phishing URLs by Varsharani Ramdas Hawanna, V.Y Kulkarni

This paper presents a unique approach to identify potentially malicious URLs used in phishing attacks. It could involve various techniques such as machine learning, data analysis, or pattern recognition to distinguish between legitimate and phishing URLs. The paper discusses the methodology, experimental results, and the effectiveness

of the proposed algorithm in accurately detecting phishing URLs. Overall, their work may contribute to enhancing cybersecurity measures by providing an efficient tool to combat phishing threats.

b) A Hybrid Model to Detect Phishing-Sites Using Supervised Learning Algorithms by

M. Amaad Ul Haq Tahir, Sohail Asghar, Ayesha Zafar, Saira Gillani

The paper introduces a hybrid model for detecting phishing sites, employing supervised learning algorithms. This approach likely combines multiple machine learning techniques to improve accuracy in identifying phishing websites. The authors may discuss the integration of various features and classifiers, such as decision trees, random forests, or neural networks, to enhance detection capabilities. The model's effectiveness is likely evaluated through experimentation, demonstrating its ability to accurately distinguish between legitimate and phishing websites.

c) Real time detection of phishing websites by Abdulghani Ali Ahmed, Nurul Amirah Abdullah

The paper introduces a method for real-time detection of phishing websites, emphasizing the critical need for timely identification and mitigation of online threats. It presents techniques or algorithms capable of swiftly analyzing web content or user behavior to flag potential phishing sites as they emerge. The authors address the challenges of real-time detection and propose solutions, which include machine learning models, heuristics, or behavioral analysis. They evaluate the effectiveness of the approach through experiments or case studies.

d) Phishing URL Detection using Machine Learning by Rutul Patel, Sanjay Kshetry ,Sanket Berad, Justin Zirthantlunga

The paper outlines a method for detecting phishing URLs using machine learning techniques. It likely describes the process of feature extraction from URLs and the creation of a classifier to differentiate between legitimate and phishing URLs. The authors probably discuss various machine learning algorithms employed for this task and evaluate.

III. PROPOSED SYSTEM

In the proposed system, we aim to detect phishing websites using a web host model based on a classification algorithm. This model will undergo training with a dedicated dataset and will be deployed. Phishing website detection will rely on URL attribute analysis. If phishing is detected, the user is alerted, safeguarding their credentials, while safe websites allow users to proceed without concern. Amalgamating diverse Neural networks, Random Forests, Decision Trees, and Support Vector Machines (SVM) are a few examples of machine learning methods. we ensure robustness and flexibility in detecting a wide range of phishing attempts. The heart of our system lies in the meticulous feature engineering process, where URL attributes are dissected to extract discriminative features indicative of phishing behaviour. Through rigorous experimentation and evaluation, we aim to identify the most effective algorithms and feature sets for the accurate and timely identification of phishing URLs.

Training Datasets

The dataset utilized in our study is sourced from Kaggle, that can be found in a particular link <https://www.kaggle.com/eswarchandt/phishing-website-detector>. This dataset comprises a collection of website URLs encompassing over 25,000 websites. Each sample within the dataset is characterized by 30 distinct website parameters, along with a corresponding class label that categorizes it as either a phishing website (labelled as 1) or a legitimate one (labelled as -1). In summary, the dataset consists of 11,054 samples, with each sample featuring 32 attributes, including the class label. This dataset offers a wealth of resources for testing and developing machine learning models for the identification of phishing websites.

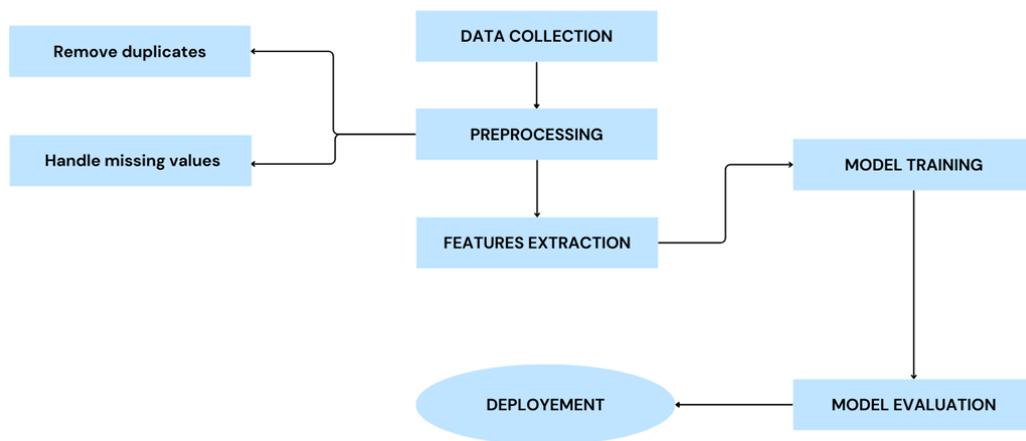


Fig 1: Flowchart of the proposed system of the model

It meticulously extracts a diverse array of attributes from the URLs within our dataset. Our approach encompasses three main categories of features: domain-based, path-based, and query-based. Domain-based features include characteristics such as the length of the domain name, the presence of hyphens within the domain, and features derived from IP addresses associated with the URLs. Path-based features focus on attributes related to the structure of the URL path, including its length and the occurrence of specific keywords that might indicate sensitive content. Additionally, query-based features involve examining parameters within the URL query string, identifying any suspicious parameters that could signal malicious intent, and considering URL encoding schemes that may obfuscate malicious payloads. By comprehensively analyzing these diverse sets of features, we aim to capture nuanced patterns and characteristics inherent in phishing URLs, thereby enhancing the effectiveness of our machine-learning models in accurately detecting phishing attempts.

Url Based Features

- a) **Domain of URL:** We scrutinize the domain portion of the URL to discern its origin and legitimacy.
- b) **Redirection '/' in URL:** Detection of double forward slashes (//) signifies potential redirection techniques employed by websites.
- c) **IP Address in URL:** Presence of IP addresses in URLs suggests direct connections to servers, vital for security monitoring.
- d) **'http/https' in Domain name:** Identification of the security protocol used within domain names is crucial for assessing website security.
- e) **'@' Symbol in URL:** Uncommon '@' symbols in URLs can indicate potential unauthorized access or phishing attempts.
- f) **Using URL Shortening Service:** Monitoring URL length and the usage of shortening services aids in usability and security analysis.
- g) **Length of URL:** Tracking URL length helps detect attempts to obscure the destination and ensures user-friendly links.
- h) **Google Index and PageRank:** Understanding a webpage's visibility, relevance, and authority aids in assessing its credibility.
- i) **Using IP:** Detection of direct IP accesses instead of domain names helps identify suspicious behaviour.
- j) **Domain Registration Length:** The duration of domain registration offers insights into website legitimacy and stability.
- k) **IframeRedirection:** Detection of webpage redirection via iframes, which can be used for malicious purposes such as phishing or malware distribution.
- l) **WebsiteForwarding:** Presence of website forwarding, where one URL automatically redirects visitors to another URL. This can obscure the true destination and may be indicative of phishing attempts.
- m) **InfoEmail:** Identification of email addresses embedded within the URL, which might be used for phishing or spamming purposes.
- n) **LinksPointingToPage:** Number of external links pointing to the webpage, which can influence the webpage's authority and credibility.
- o) **StatsReport:** Availability of a statistics report for the webpage, providing insights into its performance and activity.
- p) **NonStdPort:** Detection of non-standard ports used in the URL, which may indicate attempts to bypass standard security measures or exploit vulnerabilities.

q) ServerFormHandler: Identification of server-side form handlers, which are scripts responsible for processing form submissions on the server side. Monitoring their presence helps assess potential security risks associated with form submissions.

IV. MODEL BUILDING AND TRAINING

One of the most common and efficient forms of machine learning is supervised learning. In cases when feature-label pairs exist, supervised learning may be used to forecast a certain outcome or label from a given collection of characteristics. These feature-label pairs in order form our training set, which is utilized to build a machine-learning model. Our goal is to produce accurate forecasts for new, never-seen data. Classification and regression are the two main categories of supervised machine learning issues. Given that the suicide rate forecast is a continuous number—or, in programming parlance, a floating-point number—our data set falls under the category of regression problems. Regression is a type of supervised machine learning model that was used to train the dataset.

1. Logistic Regression: Logistic regression is a straightforward yet powerful approach for binary classification that predicts the likelihood of a binary outcome depending on one or more predictor factors. Because of its interpretability and simplicity of usage, it is frequently employed.

2. k-Nearest Neighbors (K-NN): An approach not employed in tasks involving both regression and classification. The majority class of the k nearest neighbours in the feature space is utilized to categorize new data points. While working on huge datasets, k -NN may become computationally inefficient despite the fact it is simple to learn.

3. Support Vector Classifier (SVC): Support Vector Classifier is a successful technique that may be used for problems related to regression as well as classification. It identifies what hyperplane in the feature space best separates the different categories. With the addition of kernel functions, it can handle non-linear relationships and works well with high-dimensional data.

4. Naive Bayes: Naive Bayes is a classifier with probabilities dependent on Bayes theorem, making the assumption that features in the given class label, are totally independent of each other. Actually, Naive Bayes frequently exceeds its "naive" assumption, especially when it comes to text categorizing tasks.

5. Decision Tree: It is a tree-based model that recursively partitions the features space into subsets based on the most discriminative features. Decision trees are easy to interpret and visualize, making them popular for exploratory analysis. However, they are prone to overfitting and may lack generalization.

6. Random Forest: An approach to ensemble learning that creates multiple decision trees and average or chooses among them to combine their predictions. Random Forest mitigates overfitting by combining diverse models and often yields robust performance across various datasets.

7. Gradient Boosting: It is a boosting strategy that repeatedly constructs an ensemble of weak learners (usually decision trees), with each new model aiming at reducing the errors of the before one. Gradient Boosting is well-known for its ability to handle diverse data and outstanding estimated accuracy.

8. Catboost: An optimized gradient boosting library designed for categorical features, Catboost employs various strategies to handle categorical variables effectively. It automatically deals with missing data and incorporates efficient implementations for training and inference.

9. XGBoost: Another implementation of gradient boosting, XGBoost (Extreme Gradient Boosting) is renowned for its scalability, speed, and performance. It introduces regularization techniques to control overfitting and offers flexibility in customizing the learning process.

10. Multilayer Perceptrons (MLP): It is a kind of artificial neural network character made up of multiple layers of linked neurons. Machine learning (MLP) is a popular approach for jobs like picture identification and natural language processing since it can learn intricate non-linear correlations in data and also time series prediction. They require careful tuning of hyperparameters and huge collection of data for training.

VI. RESULTS

We have tested our project with various url and go the results. Following are some of the screenshots of the results

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Multi-layer Perceptron	0.971	0.974	0.992	0.985
3	XGBoost Classifier	0.969	0.973	0.993	0.984
4	Random Forest	0.967	0.970	0.992	0.991
5	Support Vector Machine	0.964	0.968	0.980	0.965
6	Decision Tree	0.961	0.965	0.991	0.993
7	K-Nearest Neighbors	0.956	0.961	0.991	0.989
8	Logistic Regression	0.934	0.941	0.943	0.927
9	Naive Bayes Classifier	0.605	0.454	0.292	0.997

Fig 2 : Accuracy of Different models

Fig 2 shows the accuracy and various matrices of the models in different algorithms.

```

Phishing-URL-Detection-master — Python • Python app.py — 80x24
Last login: Tue Apr 23 17:26:50 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Princes-MacBook-Air:~ princekumar$ cd "Downloads/Phishing-URL-Detection-master"
(base) Princes-MacBook-Air:Phishing-URL-Detection-master princekumar$ python3 app.py
* Serving Flask app 'app' (lazy loading)
  Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
  * Debug mode: on
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000
  Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 878-762-746
  
```

Fig 3 shows the deployment status

Fig 3 shows the deployment status of the model and the running status on the debugger mode and also the deployment server status and the production status on the server with the active debugger mode.

```
127.0.0.1 - - [23/Apr/2024 17:53:21] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:21] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [23/Apr/2024 17:53:22] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/Apr/2024 17:53:28] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:28] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [23/Apr/2024 17:53:37] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:37] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [23/Apr/2024 17:53:42] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:42] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [23/Apr/2024 17:53:49] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:49] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [23/Apr/2024 17:53:58] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [23/Apr/2024 17:53:58] "GET /static/styles.css HTTP/1.1" 304 -
```

Fig 4 Response of different inputs

Fig 4 shows the response of different inputs instigated through the database through the http request from the server.



Fig 5 Web interface

Fig 5 shows the web interface for the user where the user can enter the URL to check whether the site is legitimate or not and the user can also get redirected to the legitimate website.

CONCLUSIONS

Understanding the field of machine learning models for phishing URL identification has been greatly aided by this effort. Through extensive exploration of various algorithms and thorough Exploratory Data Analysis (EDA) on the phishing dataset, we've gained a deeper understanding of the features crucial for distinguishing between safe and malicious URLs. By meticulously tuning our models, we've discerned how different parameters impact their performance, shedding light on the intricate dynamics at play. Our investigation has revealed that certain features, such as "HTTPS," "AnchorURL," and "WebsiteTraffic," significantly influence the classification of URLs as phishing or legitimate. These findings underscore the importance of scrutinizing specific attributes to effectively identify potential threats. Among the diverse array of models explored, the Gradient Boosting Classifier emerged as

particularly noteworthy, achieving an impressive classification accuracy of up to 97.4%. The comprehensive analysis conducted throughout this project not only deepens our understanding of phishing detection but also serves as a foundation for further advancements in cybersecurity research and practice.

REFERENCES

- [1] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, "Efficient deep learning techniques for the detection of phishing websites," *Sa dhana*, vol. 45, no. 1, pp. 1–18, Dec. 2020.
- [2] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics*, vol. 9, no. 9, pp. 1–24, 2021.
- [3] A. Al-Alyan and S. Al-Ahmadi, "Robust url phishing detection based on deep learning," *KSII Trans. Internet Inf. Syst.*, vol. 14, no 7, pp. 2752–2768, 2020.
- [4] J. Zhao, N. Wang, Q. Ma, and Z. Cheng, "Classifying malicious URLs using gated recurrent neural networks," in *Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, vol. 773, 2020, pp. 385–394.
- [5] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 649–657.
- [6] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 1746–1751.
- [7] K.L.Chiew, C.L.Tan, K. Wong, K.S.C. Yong, and W.K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci.*, vol. 484, pp. 153–166, May 2022.