Phishguard: ML- based phishing detection system

Bhagyashree Satarkar[1], Yasar Sayyad[2], Dhavalsinh Vibhute[3], Vishal Wadgaonkar[4],

[1234]Department of computer engineering,Genba Sopanrao Moze college of engineering,pune

The phishing attacks is a threat based on the fraudulent communication which has grown so much in the recent years. It is a type of attack where an attacker deceives the user by sending fake messages using various social media platforms or E-mail. The attackers steals the user's information and important data. They are hard to detect because attackers can design the incoming messages looks like from a legitimate User. It is necessary to protect the users from such phishing attacks before getting caught into any hazardous trouble situation.

Keywords: Machine Learning, Uniform Resource Locator (URL), logistic regression, support vector classifier, random forest classifier, multinomial naive bayes.

1.Introduction

Phishing attacks are cybercrime using social engineering to deceive users into stealing their information, such as personal identity, financial information, etc. Masquerading as legitimate sources, attackers can reach victims by sending fraudulent messages using emails (such as Gmail, Outlook, etc.) or social media platforms (like Twitter, Facebook, etc.). Users become vulnerable if they input their information or download attachment files. Phishing attacks have become a global threat due their expanded extremely fast expansion in the most recent couple of years 5–8. It is absurd to expect a 100% phishing attack detection approach, as attackers routinely change their attacking methods. As such, various solutions have been suggested by experts over the previous years to detect and mitigate phishing attacks. However, the burden of phishing attacks still exists, and developing an efficient anti-phishing approach has become challenging.

Moreover, most anti-phishing solutions produce high false positives and are not capable of dealing with zero-hour attack. Email is the mainstream which attackers use to deploy phishing attacks. In addition, messaging has now bought into the mainstream in delivering phishing attacks. Phishing approaches are usually separated into two groups: user awareness and a systematic approach.

Machine learning has become a critical tool in the fight against phishing on the web. By analyzing large volumes of data, machine learning algorithms can identify patterns and

anomalies associated with phishing attacks. These algorithms can learn characteristics related to URL structure, domain age, website content, and email metadata to detect phishing attempts with high precision. Thus, using machine learning not only enhances detection capabilities but also reduces false positives, thereby avoiding the misidentification of legitimate websites and emails as malware.

2.Background

Machine learning and data-driven approaches have been increasingly employed to solve cybersecurity-related problems. The phishing detection research landscape shows that, through natural language processing techniques, robust results have been obtained. Most of this research is centered on how to extract, from the text and the metadata of the e-mail, highly distinctive features that allow it to identify differences and similarities among these messages, in order to separate them in phishing or legitimate e-mails.

One of the first approaches to phishing e-mail detection based on machine learning was proposed by Fette et al. It generated features based on e-mail texts and properties, such as if these e-mails contain javascript code, the number of links in the e-mail, or the number of dots in the present Uniform Resource Locators (URLs). It detected over 96% of the phishing e-mails when submitting the best ten features they found to the Random Forest classification algorithm.

The further improvement and enhancement across many years till date.

Machine-Learning-Based Phishing Identification Systems

Machine learning is increasingly popular for phishing detection because of its ability to classify new and unknown threats. ML algorithms learn from datasets containing legitimate and phishing website attributes, enabling them to identify patterns and distinguish between the two categories.

Early Research: One notable system, CANTINA, used a text-classification approach to identify phishing websites by analyzing term frequency-inverse document frequency (TF-IDF) of keywords. While effective, this method was limited to English vocabulary and prone to false positives. An improved system, CANTINA+, incorporated additional HTML attributes to enhance accuracy, achieving a detection rate of 92% but still faced challenges with false-positive predictions.

Advanced Approaches: Recent studies have explored a variety of ML techniques. For instance, an ensemble learning system using weighted confidence and adaptive regularization outperformed traditional methods, achieving high accuracy with minimal resource usage. Another study utilized Apriori rule-mining algorithms to identify phishing URLs based on attributes like URL length, subdomain usage, and special characters, achieving 93% accuracy.

Innovative Techniques: Recent innovations include metaheuristic algorithms, natural language processing (NLP)-based systems, and neural networks. For example, NLP techniques analyze semantic patterns in URLs and website content to detect phishing attempts, achieving high precision rates. Self-structuring neural networks have also been applied to phishing detection, with some models achieving over 97% accuracy by analyzing domain-based, address bar-based, and HTML features.

3.Methods and Requirements

The study for the phishing detection system is focused on the URL classification using machine

learning algorithms. Cybercrimes are growing with the growth of Internet architecture worldwide, which needs to provide a security mechanism to prevent an attacker from getting confidential content by breaching the network through fake and malicious URLs. A phishing dataset was used to perform the experiments.

3.1 Phishing URL Dataset:

The dataset used in this project is the SMS Spam Collection dataset, which contains 5,572 SMS messages. Each message is labeled as either spam or ham (not spam). The dataset has a slight imbalance, with 13% of the messages labeled as spam. The dataset used for this study was obtained from a well-known repository, Kaggle, which provides benchmark datasets for research purposes. It included 11,054 records from over 11,000 websites, with attributes indicative of phishing or legitimate URLs.

3.2 Data Preprocessing Techniques:

Before taking the data into the machine learning (ML) model several preprocessing steps were carried out on the dataset. The data preprocessing technique is a critical step in the machine learning. It involves following steps which we require and carried out:

Tokenization: Splitting the message into individual words or tokens.

Lowercasing: Converting all characters to lowercase to ensure uniformity.

Stopword Removal: Removing common words like “is,” “the,” “and,” etc., that don’t contribute much to the classification.

Lemmatization: Converting words to their base form (e.g., "running" to "run") to reduce the feature space.

3.3 Data Augmentation:

Data augmentation techniques are critical for improving the performance of machine learning models, especially when dealing with limited data. These techniques artificially expand the training dataset by generating new, transformed versions of the original data. This helps to improve model generalization, reduce overfitting, and increase the robustness of the model.

To address the class imbalance, data augmentation techniques were used to artificially increase the number of spam messages by adding slight variations to the existing messages. This helped prevent the model from becoming biased towards the majority class (ham).

In the natural language processing the text data augmentation was executed on the dataset. such as synonym replacement, random insertion, random deletion, text shuffling, word embedding perturbation, etc.

3.4 Model validation:

To develop and validate the models, we randomly partitioned the data into training and test sets,

The dataset was split into 80% training and 20% testing sets. The training data was used to build the model, while the test data was used to evaluate its performance.

As the model validation is a critical part of the machine learning process, helping assess how well a model generalizes to unseen data. It ensures that a model is not overfitting to the training data and is capable of making accurate predictions on new, unseen data.

In the Train Test-Split, we split the data into the 4:1 ratio as mentioned above. It is a simplest form of model validation which helps in training and testing the data.

The only disadvantage is that he model's performance might be overly dependent on the specific split of the data.

3.5 Algorithm

3.5.1 Naive Bayes Algorithm:

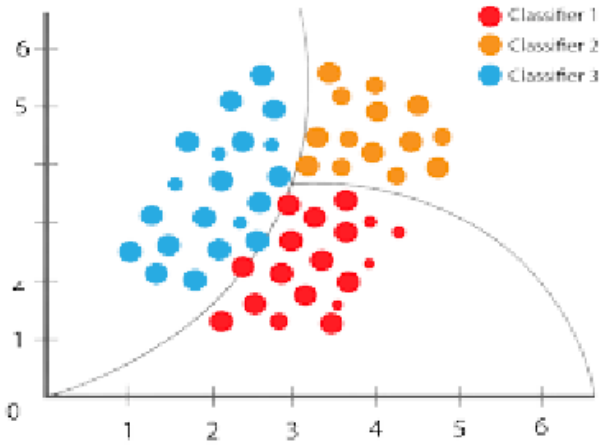
After evaluating several algorithms, the Naive Bayes classifier was selected for its simplicity and efficiency in text classification tasks. Other models, such as SVM and Random Forest, were considered, but Naive Bayes provided the best trade-off between accuracy and computational cost. The Naive Bayes classifier is based on Bayes' theorem, which calculates the probability of a message being spam given the presence of specific words. The model assumes that the presence of each word in the message is independent of the other words, an assumption that simplifies computation but often performs surprisingly well in practice.

The foundation of Naive Bayes is Bayes Theorem, expressed as:

P(A|B) = (P(B|A) ∗ P(A)) / P(B)

Where,

A is the class and B is the feature vector. The naive Bayes algorithm is a probabilistic classifier built on Bayes' theorem. P (B|A), P(A), and P(B) are the probabilities measured from earlier known instances, such as training data. Classification errors are minimized by selecting a class that maximizes the probability P(A|B) for every occurrence.



(knive bayes algorithm)

Training is quick because it involves calculating probabilities for each feature in the knive bayes algorithm. Performs well even with small datasets, provided the independence assumption is reasonably accurate.

3.5.2Support Vector Classifier:

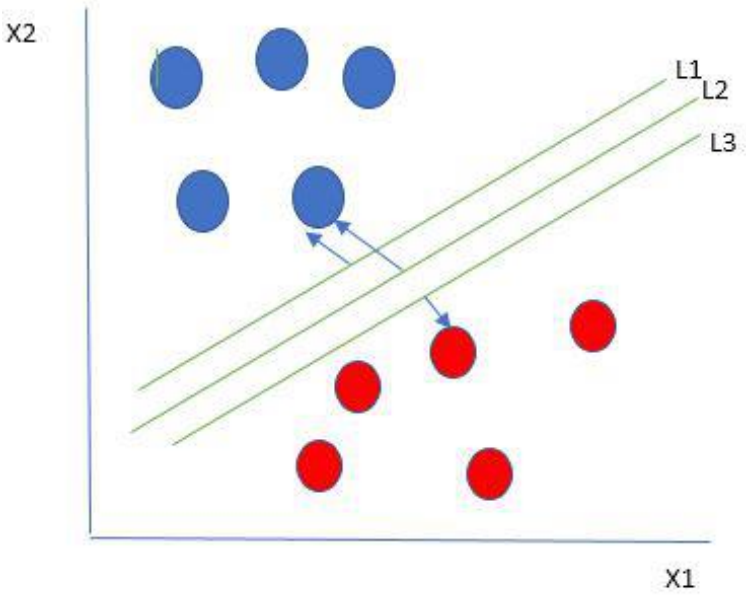
SVC is a powerful classification algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate different classes. The key objective is to maximize the margin between data points of different classes, with support vectors being the data points closest to the hyperplane. it is mostly used in classification problems, where it provides the best accuracy between two classes.

In SVM, a hyperplane is a decision boundary that separates data points of different classes. In 2D, this is a line; in 3D, it's a plane; and in higher dimensions, it is a hyperplane.

Margin-The margin is the distance between the hyperplane and the nearest data points from either class. The goal of SVM is to maximize this margin, which helps the model generalize better to unseen data.

Support vectors-These are the data points that lie closest to the hyperplane and play a crucial role in determining the optimal hyperplane.

Kernel trick- SVM can be extended to handle non-linear data by using a kernel function, such as the radial basis function (RBF), polynomial kernel, or linear kernel. This allows SVM to map data to higher dimensions, making it easier to find a separating hyperplane.



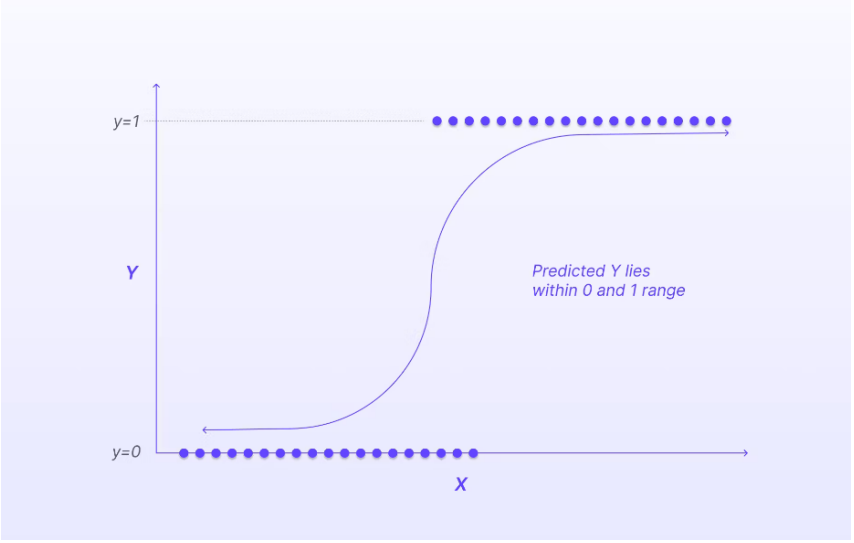
(Support vector Classifier)

3.5.3Logistic Regression:

Logistic Regression is a statistical model used for binary classification tasks, where the goal is to predict the probability that an instance belongs to a particular class. Despite its name, it is a classification algorithm, not a regression algorithm. Logistic Regression is used for predicting the probability of a binary outcome based on one or more input variables.
It applies a logistic function to model a binary dependent variable. The algorithm estimates the probability that a given instance belongs to a particular class, with the output ranging between 0 and 1.

σ(z)=1/1+(e−z)

The logistic regression is best for problems where the decision boundary is linear. Also it is built in with L1 and L2 regularization.
The computation complexity varies from the low to the moderate level, not too high.



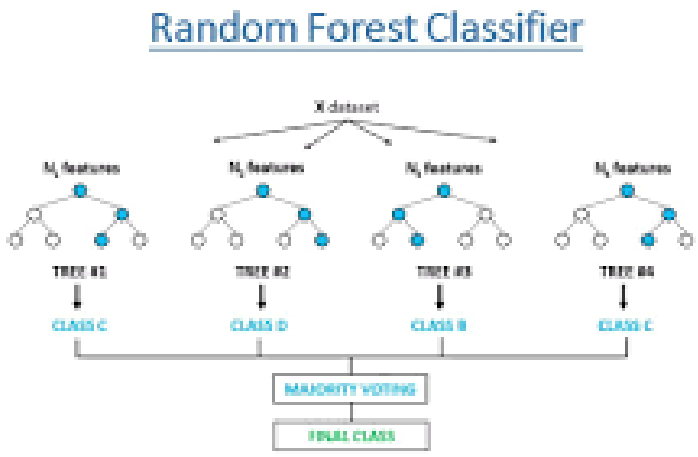
(logistic regression)

Logistic regression models can be regularized to prevent overfitting by penalizing large coefficients using L1 and L2.
Unlike more complex models (e.g., Random Forest, SVM), logistic regression has very few hyperparameters to tune..
Regularization parameters like **L1 (Lasso)** or **L2 (Ridge)** regularization are optional and straightforward to incorporate.

3.5.4 Random Forest:

Random Forest is an ensemble learning method primarily used for classification and regression tasks. It combines multiple decision trees to create a more robust, accurate model that reduces the risk of overfitting. Each tree in the forest is trained on a random subset of the data, and the final prediction is made based on the majority vote (in classification) or the average (in regression) of the individual trees. This method is an ensemble method that is better than a single decision tree because it reduces over fitting by averaging the result.

The random forest classifier uses a decision tree as the base classifier. A decision tree is a flowchart-like structure where each node represents a decision based on a feature, and each leaf node represents an outcome (class label or continuous value). It is prone to overfitting when used alone, especially with noisy data.



(random forest classifier)

Random Forest often outperforms single decision trees and many other algorithms due to its ensemble approach, reducing overfitting and improving accuracy.
By averaging the predictions of multiple trees, Random Forest minimizes the risk of overfitting, especially with large datasets.
Random Forest can model complex, nonlinear relationships between features and the target variable.
Since it combines multiple trees, individual noisy samples or outliers have minimal impact on the overall model performance.
Provides an internal performance estimate using samples not included in the bootstrap (OOB samples), reducing the need for a separate validation set.
Trees in a Random Forest are built independently, allowing for easy parallelization and faster training on multicore processors.

4 Discussion

There are different machine learning algorithms used while performing the study of the project
Comparative analyses of all the multiple machine learning models. presented the clear and significant effects of machine learning models in this study.
The comparative analyses illustrate that the machine learning model that consists of linear approaches or probabilistic approaches, such as linear regression and support vector machines, do not perform very well and show very low results.
Instead if we use the logistic regression algorithm it performs very well and shows very efficient results.
In addition, the conclusions of most studies are based on tiny datasets and cannot be applied to larger populations. Using large, this study presents a hybrid model for phishing detection prediction that overcomes these constraints and achieves a greater level of accuracy.
The proposed method achieves 98.12% accuracy.

5 Conclusion

The Internet is becoming an indispensable part of the modern world, continuing to expand at an unprecedented rate. However, alongside its growth, cybercrimes are increasing rapidly, with malicious and suspicious URLs posing a major threat.
These harmful URLs significantly impact the reliability of internet services and the operations of industrial companies. Ensuring privacy and confidentiality has become a critical challenge in this digital era.
Cyber attackers often exploit phishing emails or URLs to bypass security measures and infiltrate secure networks. Phishing URLs are crafted to mimic legitimate ones, making them an easy and effective tool for unauthorized access to sensitive or private networks.
A dataset consisting of 32 URL attributes and more than 11054 URLs was extracted from 11000+ websites. This dataset was extracted from the Kaggle repository and used as a benchmark for research. This dataset has already been presented in the form of vectors used in machine learning models.
Logistic regression, Knive bayes algorithm,SVC, random forest algorithm were applied to perform the experiments and achieve the highest performance results.

6 Reference

1. N. Z. Harun, N. Jaffar, and P. S. J. Kassim, ‘‘Physical attributes significant in preserving the social sustainability of the traditional malay settlement,’’ in Reframing the Vernacular: Politics, Semiotics, and Representation. Springer, 2020, pp. 225–238.
2. D. M. Divakaran and A. Oest, ‘‘Phishing detection leveraging machine learning and deep learning: A review,’’ 2022, arXiv:2205.07411.
3. A. Akanchha, ‘‘Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates,’’ Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.
4. H. Shahriar and S. Nimmagadda, ‘‘Network intrusion detection for TCP/IP packets with machine learning techniques,’’ in Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Cham, Switzerland: Springer, 2020, pp. 231–247.
5. J. Kline, E. Oakes, and P. Barford, ‘‘A URL-based analysis of WWW structure and dynamics,’’ in Proc. Netw. Traffic Meas. Anal. Conf. (TMA), Jun. 2019, p. 800.
6. A. K. Murthy and Suresha, ‘‘XML URL classification based on their semantic structure orientation for web mining.
7. A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ‘‘Phishing website detection: An improved accuracy through feature selection and ensemble learning,’’ Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019.
8. A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, ‘‘PhishAri: Automatic realtime phishing detection on Twitter,’’ in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.
9. S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security— ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.
10. George and P. Vinod, ‘‘Composite email features for spam identification,’’ in Cyber Security. Singapore: Springer, 2018, pp. 281–289.
11. S. Hota, A. K. Shrivas, and R. Hota, ‘‘An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique,’’ Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.
12. G. Sonowal and K. S. Kuppusamy, ‘‘PhiDMA—A phishing detection model with multi-filter approach,’’ J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020
13. M. Zouina and B. Outtaj, ‘‘A novel lightweight URL phishing detection system using SVM and similarity index,’’ Hum.-Centric Comput. Inf. Sci., vol. 7, no. 1, p. 17, Jun. 2017.
14. R. Ø. Skotnes, ‘‘Management commitment and awareness creation—ICT safety and security in electric power supply network companies,’’ Inf. Comput. Secur., vol. 23, no. 3, pp. 302–316, Jul. 2015.
15. R. Prasad and V. Rohokale, ‘‘Cyber threats and attack overview,’’ in Cyber Security: The Lifeline of Information and Communication Technology. Cham, Switzerland: Springer, 2020, pp. 15–31.
16. T. Nathezhtha, D. Sangeetha, and V. Vaidehi, ‘‘WC-PAD: Web crawling based phishing attack detection,’’ in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2019, pp. 1–6.

17. R. Jenni and S. Shankar, ‘‘Review of various methods for phishing detection,’’ EAI Endorsed Trans. Energy Web, vol. 5, no. 20, Sep. 2018, Art. no. 155746.
18. S. Bell and P. Komisarczuk, ‘‘An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,’’ in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY, USA: Association for Computing.
19. A. K. Jain and B. Gupta, ‘‘PHISH-SAFE: URL features-based phishing detection system using machine learning,’’ in Cyber Security. Switzerland: Springer, 2018, pp. 467–474.
20. S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, ‘‘Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions,’’ in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., Apr. 2010, pp. 373–382.

1.