Machine Learning-Based Phishing Detection from URL using Support Vector Machines (SVMs) and Random Forests

Aung Khant Kyaw (Student ID - 1583883), Jiqiang Wang (Student ID - 1564710)
School of Computing
Unitec Institute of Technology
Auckland

April 4, 2024

Contents

1 Abstract

Phishing attacks remain a critical threat in the cybersecurity landscape, exploiting human vulnerabilities to trick individuals into surrendering sensitive information, leading to financial losses and reputational damage [18]. This project proposes a machine learning-based approach for detecting phishing URLs, employing Support Vector Machines (SVMs) and Random Forests models. The objective is to classify URLs as legitimate or malicious based on features extracted from their structure. SVMs and Random Forests are well-suited for this task due to their effectiveness in binary classification tasks, their ability to handle high-dimensional data, and their resistance to

overfitting [1], [2]. Publicly available labeled datasets, such as UCI Machine Learning Repository's Phishing URL (PhiUSIIL) dataset [3], will be utilized to train and evaluate the models. By achieving high accuracy in phishing URL identification, this project aspires to contribute to the fortification of cybersecurity measures and mitigate the financial and reputational damage caused by phishing attacks [4].

2 Methodology

2.1 Preferred Learning Methods

This project adopts two powerful supervised learning techniques to classify phishing URLs: Support Vector Machines (SVMs) and Random Forests.

SVMs are adept at identifying hyperplanes that effectively separate data points belonging to different classes [4]. In our context, SVMs will discern between legitimate and phishing URLs based on extracted features, showcasing resilience in handling high-dimensional data and addressing imbalanced datasets commonly encountered in phishing detection, where phishing URLs are typically the minority class [1]. On the other hand, Random Forests utilizes an ensemble approach, combining multiple decision trees to create a robust and accurate classifier [2]. Each tree in the forest is trained on a unique subset of features and data points, resulting in a diverse set of URL classification rules. This diversity mitigates the risk of overfitting and enhances the model's generalizability to new, unseen data instances [2]. Thus, by leveraging the strengths of SVMs and Random Forests, this project aims to develop a comprehensive phishing URL detection system capable of accurately identifying and mitigating potential cyber threats.

However, SVMs may struggle with large datasets due to computational intensity. They may require careful kernel choice and parameter tuning to ensure optimal performance [1], [4] At the same time, Random Forests can suffer from biases if the underlying decision trees are not adequately diversified and may not be as interpretable as SVMs due to their ensemble nature [2], [5]. These methods cover each other's limitations: SVM's boundary clarity can be reinforced by the Random Forest's diverse decision perspectives, providing a robust classification strategy. While SVMs offer precision, Random Forests contribute breadth in decision-making, ensuring our detection system remains accurate and generalizable [2], [4].

2.2 Datasets Required for the Project

The project will leverage publicly available labeled URL datasets to train and evaluate the machine learning models.

UCI Machine Learning Repository: The UCI Machine Learning Repository offers a rich collection of datasets, including Phishing URL (Website) (PhiUSIIL), which contains features and labels for both phishing and legitimate URLs. This dataset, which consists of 235,795 instances with a balanced mix of 134,850 legitimate and 100,945 phishing URLs and includes 54 comprehensive features extracted from the URLs and webpage content, such as 'CharContinuationRate', 'URLTitleMatchScore', and 'TLDLegitimateProb' [3] will be utilized for training and testing the models.

2.3 Justification of Learning Methods

This project focuses on Support Vector Machines (SVMs) and Random Forests due to their distinctive strengths in phishing URL detection tasks:

Support Vector Machines

- High-Dimensional Data Handling: Phishing URL detection involves a multitude of features like lexical characteristics, domain information, and suspicious characters. SVMs excel at handling such high-dimensional data by identifying the optimal hyperplane that effectively separates legitimate and phishing URLs within the feature space. This makes them adept at dealing with the intricate feature representations in phishing datasets [4].

- Class Imbalance Robustness: Phishing datasets often exhibit class imbalance, where phishing URLs are outnumbered by legitimate ones. SVMs are known to perform well in such scenarios. They prioritize maximizing the margin between classes rather than simply minimizing overall classification error [1]. This ensures the model isn't biased towards the majority class, maintaining its accuracy in detecting phishing URLs.

Random Forests:

- Feature Importance Insights: Random Forests offer a unique benefit by providing insights into feature importance. They construct multiple decision trees and combine their predictions. This ensemble approach allows Random Forests to identify the features that most significantly contribute to classification, revealing the key characteristics that differentiate phishing URLs from legitimate ones [5].

- Overfitting Reduction: Random Forests effectively reduce the risk of overfitting, a common concern with complex datasets. By aggregating predictions from multiple trees, Random Forests lessen the tendency of individual trees to memorize the training data. This enhances the model's ability to generalize to unseen URLs, which is crucial for real-world performance where the model encounters new, unseen instances [6].

Both SVMs and Random Forests possess unique strengths that make them well-suited for phishing URL detection. SVMs excel in handling high-dimensional and imbalanced data, while Random Forests offer insights into feature importance and mitigate overfitting. By leveraging the strengths of both algorithms, this project aims to build robust and accurate phishing URL detection systems.

2.4 Project Implementation

Data Preprocessing: Data preprocessing is a crucial step in any machine learning project as it lays the foundation for accurate model training and evaluation. It involves cleaning and preparing the raw data before feeding it into machine learning algorithms. In our project, we begin by loading the dataset from the provided CSV file using the pandas library [7] a powerful data manipulation tool in Python. This allows us to easily manipulate and analyze the data, including handling missing values by either imputing appropriate values or removing entries with too many missing values, depending on the dataset's characteristics [8]. Additionally, leveraging Python's string manipulation capabilities, we clean the data by removing duplicates, converting text to lowercase for consistency, and handling any encoding issues that may arise [9].

Feature Engineering: Feature engineering plays a vital role in extracting meaningful information from raw data, thereby improving the performance of machine learning models. In our project, using Python's extensive libraries such as pandas and NumPy, we extract relevant features from the dataset that are indicative of phishing URLs. These features may include URL length, domain length, presence of special characters in the URL, number of subdomains, presence of HTTPS protocol, and various other indicators of phishing activity [10]. By harnessing Python's flexibility and ease of use, we carefully select and transform these features, providing valuable input to the machine learning models and enabling them to better discriminate between legitimate and phishing URLs.

Model Training and Evaluation: Once the data is preprocessed and the features are engineered, we proceed to train and evaluate machine learning models for phishing URL detection using Python. We split the data into training and testing sets, typically using an 80-20 split, to assess the model's performance on unseen data [11]. Leveraging Python's scikit-learn library, we train Support Vector Machines (SVM) and Random Forest models on the training data, as these algorithms are commonly used for binary classification tasks like phishing URL detection [11]. To optimize the models' performance, we perform hyperparameter tuning, adjusting the model parameters to achieve the best possible results [12]. Finally, we evaluate the models using various metrics such as accuracy, precision, recall, and F1-score on the testing data [11].

Model Comparison and Selection: After evaluating the models, we compare their performance based on the evaluation metrics to determine which one is better suited for phishing URL detection [11]. Leveraging Python's data visualization libraries such as matplotlib or seaborn, we analyze the models' accuracy, precision, recall, and F1-score and selecting the one that achieves superior performance. By leveraging Python's rich ecosystem of tools and libraries, we carefully considering the strengths and weaknesses of each model, ensuring that our phishing URL detection system is both accurate and reliable.

Deployment and Testing: Once the best-performing model is selected, we proceed to deploy it as a standalone phishing URL checker tool for users [13]. Leveraging Python-based web frameworks like Flask or Django, we create a user-friendly interface that allows users to input URLs and receive predictions on whether they are legitimate or phishing. Additionally, we continuously monitor and evaluate the model's performance on new data to ensure its effectiveness over time [14]. Phishing tactics evolve rapidly, so periodically retraining the model with fresh data is necessary to keep it up-to-date and capable of accurately detecting new threats. Python's versatility and ease of integration make it an ideal choice for both model development and deployment, ensuring that our phishing URL detection system remains robust and effective in real-world scenarios.

Summary

The project proposes a comprehensive strategy to combat phishing attacks using machine learning, specifically SVMs and Random Forests. Phishing attacks remain a significant cybersecurity threat, exploiting human vulnerabilities for financial and reputational damage. This system aims to detect phishing URLs by training on labeled datasets like the PhiUSIIL dataset from the UCI Machine Learning Repository. SVMs and Random Forests are chosen for their effectiveness in binary classification, handling high-dimensional data, and resisting overfitting. The project involves data preprocessing, feature engineering, model training, and evaluation using Python libraries like pandas and sci-kit-learn. For deployment, frameworks like Flask or Django will be utilized. The goal is to enhance cybersecurity by providing accurate phishing protection, thereby mitigating financial and reputational harm caused by such attacks.

3 References

[1] O. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 01 2019.

[2] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and M. S. A. N., "Phishing detection using random forest, svm and neural network with backpropagation," in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, 2020, pp. 391–394.

[3] A. Prasad and S. Chandra, "PhiUSIIL Phishing URL (Website)," UCI Machine Learning Repository, 2024, dOI: https://doi.org/10.1016/j.cose.2023.103545.

[4] A. Aljofey, S. Yousuf, and D. Whalen, "An effective detection approach for phishing websites using url and html features," *Nature Scientific Reports*, vol. 12, no. 1, p. 10841, 2022. [Online]. Available: https://www.nature.com/articles/s41598-022-10841-5

[5] Y. Li, L. Sun, J. Wu, and Y. Zhang, "A survey on machine learning techniques for phishing detection," *Artificial Intelligence Review*, pp. 1–22, 2023. [Online]. Available: https://www.researchgate.net/publication/354069207_A_Survey_of_Machine_Learning-Based_Solutions_for_Phishing_Website_Detection

[6] L. Xiao, Y. Liu, and X. Luo, "Cost-sensitive random forests for imbalanced class learning in phishing website detection," *IEEE Access*, vol. 8, pp. 84 433–84 443, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8252051

[7] W. McKinney, "pandas: a foundational python library for data analysis and statistics," *Python for High Performance and Scientific Computing*, vol. 14, 2011.

[8] T. Hernandez, "Handling missing data with python," *Journal of Open Source Software*, vol. 3, no. 22, p. 547, 2018.

[9] J. VanderPlas, *Python Data Science Handbook*. O'Reilly Media, 2016.

[10] J. Brownlee, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Machine Learning Mastery, 2020.

[11] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Packt Publishing, 2019.

[12] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[13] N. Gift, J. Jones, and B. F. Hilburn, *Pragmatic AI: An Introduction to Cloud-Based Machine Learning*. Addison-Wesley Professional, 2020.

[14] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.