International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 08 | Aug - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Recognition of Malicious URLs Using Machine Learning

Dr. Sweety Garg [1], Ms. Sayed Shifa Mohd Imran[2]

[1]Assistant Professor, Department of Computer and Information Science, Nagindas Khandwala College, Mumbai, Maharashtra, India

[2] Student, Department of MSc.IT, Nagindas Khandwala College, Mumbai, Maharashtra, India

Abstract

The surge in online activities has heightened the risk of cyber-attacks, with phishing emerging as a notably widespread threat. Conventional phishing detection methods, such as blacklists and heuristic analysis, fall short when it comes to identifying novel and complex phishing schemes. To tackle this issue, we propose a machine learning-based solution for detecting phishing websites. Our model utilizes a combination of features—such as URL structure, domain attributes, and content characteristics—to categorize websites as either phishing or legitimate. The model's outstanding overall accuracy of 96.9% underscores the need to evaluate both precision and recall when determining the effectiveness of machine learning models. The results of this study are significant for developing robust URL classification models and advancing the field of cybersecurity. Future research could aim to improve the model's performance by integrating more features, developing real-time phishing detection systems, and exploring new attributes that can be derived from URLs.

Keywords: Phishing, URL, Machine Learning, Classification, Detection

Introduction

In the digital era, the internet is vital for communication, commerce, and information sharing. However, this widespread connectivity also brings risks, particularly from cybercriminals who exploit online platforms to commit fraud. Phishing, a common and deceptive tactic, involves creating fraudulent websites are used to steal sensitive information such as passwords and credit card details. These counterfeit sites are designed to closely resemble legitimate ones, making them challenging to identify without specialized tools.

Conventional phishing detection methods usually depend on blacklists of known malicious URLs or heuristic techniques that analyze website content for suspicious features. While these methods can be useful, they are constrained by predefined lists and static rules, which may not effectively capture emerging or sophisticated phishing schemes. Moreover, the vast number of websites and the rapid evolution of phishing tactics present significant challenges for these traditional approaches.

To overcome these limitations, there is increasing interest in using machine learning (ML) techniques for phishing detection. ML offers a dynamic approach to identifying malicious URLs by analyzing patterns and features from large datasets. Unlike traditional approaches, machine learning models can continuously learn and adapt to emerging phishing strategies by training on varied and extensive datasets. This ability to generalize from historical data enhances the accuracy of detecting previously unseen phishing attempts.

This paper explores the use of machine learning for detecting malicious URLs, with an emphasis on developing a robust model capable of distinguishing phishing sites from legitimate ones using features like URL structure, domain attributes, and content characteristics. By utilizing advanced algorithms and

data-driven approaches, the study seeks to enhance the accuracy and efficiency of phishing detection systems, offering a scalable solution to safeguard users against evolving online threats and contributing to advancements in cybersecurity.

Literature Review

Mohammad et al. (2014) discussed the effectiveness of various machine learning algorithms in detecting phishing websites. Their study highlights that algorithms such as Naïve Bayes and Support Vector Machines (SVM) show high accuracy in classification tasks. They emphasized the importance of feature extraction, noting that URL lexical features and domain-related information significantly contribute to the detection process.

Abdelhamid et al.(2014) focused on using associative classification in phishing detection. Their model, which integrates URL and HTML content features, demonstrates high accuracy and interpretability. The study suggests that associative classification, combined with other machine learning techniques, can improve phishing detection systems.

Jain and Gupta (2017) provide a comprehensive survey of phishing detection techniques using machine learning. They categorize features into URL-based, content-based, and behavior-based, and conclude that combining these features enhances the detection accuracy. The study finds that decision tree-based models and ensemble methods like Random Forest and Gradient Boosting are particularly effective.

Patil and Patil (2020) explore phishing detection using machine learning models such as logistic regression, decision trees, and neural networks. They emphasize the use of hybrid features, including URL patterns and host-based features. Their findings indicate that hybrid models significantly outperform single-feature-based models in detecting phishing websites.

Marchal et al. (2014) propose a real-time phishing detection system leveraging machine learning. The study utilizes features like domain attributes, URL length, and suspicious tokens, demonstrating high accuracy and low false-positive rates. Their work highlights the potential of machine learning for real-time phishing detection in practical scenarios.

Research Objectives

1. To create a machine learning model capable of accurately detecting phishing websites with high precision.

2. To classify websites as either phishing or non-phishing based on a range of features and characteristics.

Methodology

The rise in internet usage has led to an increase in malicious URLs, posing serious threats to computer security and causing substantial harm. Traditional URL filtering techniques, such as rule-based systems and blacklists, often struggle to detect new and evolving malicious URLs effectively. Machine learning (ML) approaches offer a more adaptive solution, capable of learning from extensive datasets and adjusting to emerging patterns in malicious URLs. This study utilizes a dataset from Kaggle to evaluate the performance of a machine learning model in detecting malicious URLs. The Confusion Matrix is employed as a tool for visualizing the model's predictions compared to the actual labels.

a) Sample data

index	url	label
29374	eco-essence.cl/yh/yah	bad
66543	crowle.org/?p=179	good
177758	amazon.com/20th-Century-Fox-Years-Great/dp/B0049IRXCC	good
307892	news1130.com/sports/article/160241--tories-hope-larry-smith-s-star-appeal-will-help-them-crack-the-montreal-area	good
298941	myarchives.net/showthread.php?2390-The-Best-10-Bustiest-Playmates-of-All-Times	good
18404	redcelloproductions.com/waplog/com.htm	bad
291654	metacafe.com/watch/6858926/battleship_movie_trailer/	good
213491	corporationwiki.com/Massachusetts/Boston/carl-s-sloane/52050498.aspx	good
389478	wwnh6.hikonefabebppd.com/51coec5tl3\nwww.blogbuilderbasics.com/95d9497e546917a8e3d6bb9035e7f633.php?q=b66e11b4180154c23ba07188a51a4c07	bad
150807	theleek.net/	good

Table 1: Sample Dataset

b) Libraries used

1. Pandas: For manipulating and analyzing data.
2. Scikit-learn: Provides various machine learning algorithms for classification, regression, clustering, etc.
3. TfidfVectorizer: Transforms text data into numerical features using the TF-IDF method.
4. LogisticRegression: Implements logistic regression, a common classification algorithm.
5. DecisionTreeClassifier: Implements decision tree classification.
6. train_test_split: Divides data into training and testing sets.
7. accuracy_score: Calculates model accuracy.
8. classification_report: Generates a report including precision, recall, and F1-score.
9. Seaborn: A visualization library built on matplotlib.
10. Matplotlib: For plotting graphs.

c) Implemented methods

The proposed approach for detecting malicious URLs involves two main steps: Feature Extraction and Decision Tree Classification.

Step 1: Feature Extraction

This phase involves extracting text-based features from URLs. The TF-IDF vectorization technique quantifies the significance of each word within a URL. TF-IDF, a common method in natural language processing, assesses word importance by combining its frequency with its inverse document frequency.

The TF-IDF vectorization process includes:

1. Tokenization: Splitting URLs into individual tokens or words.
2. Frequency Calculation: Counting occurrences of each token in the URL.
3. Inverse Document Frequency Calculation: Calculating the inverse document frequency by dividing the total number of documents by the number of documents that contain the token.
4. TF-IDF Calculation: Multiplying the token frequency by its inverse document frequency to derive the TF-IDF score.

   The output is a TF-IDF vector representing the URL's text-based features.

Step 2: Decision Tree Classification

This step involves training a decision tree classifier using the TF-IDF vectors and their respective labels (malicious or benign). The training process using the scikit-learn library involves:

1. Preparing Training Data: Splitting data into input features (TF-IDF vectors) and output labels (malicious or benign).
2. Model Training: Training the decision tree classifier with the training dataset.
3. Model Evaluation: Evaluating the model's performance using test data.

The decision tree classifier uses recursive partitioning to classify URLs based on their TF-IDF vectors, forming a tree-like structure by partitioning data based on the most significant features.

The study includes the following methods:

1. TF-IDF Vectorization: For extracting features from URLs.
2. Decision Tree Classification: For training a classifier with TF-IDF vectors and labels.
3. Model Evaluation: To assess the performance using test data.
4. Feature Selection: To identify the most informative features from TF-IDF vectors.

Results

The machine learning model shows high accuracy and precision in distinguishing between malicious and benign URLs. With an accuracy of 96.9%, the model correctly classified nearly 97% of URLs. Precision and recall for the "malicious" class are 0.94 and 0.89, respectively, indicating strong performance

in identifying true positives but some susceptibility to false negatives. In contrast, precision and recall for the "benign" class are 0.98 and 0.99, respectively, reflecting high accuracy in identifying true negatives. The overall F1-score of 0.95 indicates strong model performance.

The classification report includes macro and weighted average metrics, with a macro average F1-score of 0.96, suggesting effective identification of both classes, and a weighted average F1-score of 0.97, indicating better accuracy in identifying benign URLs.

```
Accuracy: 0.9691056330491242
Classification Report:
              precision    recall  f1-score   support

         bad       0.94      0.89      0.91     14964
        good       0.98      0.99      0.98     69129

    accuracy                           0.97     84093
   macro avg       0.96      0.94      0.95     84093
weighted avg       0.97      0.97      0.97     84093
```

Figure 1: Classification Report

The confusion matrix further illustrates the model's performance, showing accurate classification of 13,259 malicious URLs and 68,236 benign URLs, with misclassifications of 1,705 benign URLs as malicious and 893 malicious URLs as benign. The overall accuracy of 84,093 URLs highlights the model's effectiveness, though fine-tuning is necessary to minimize false positives.

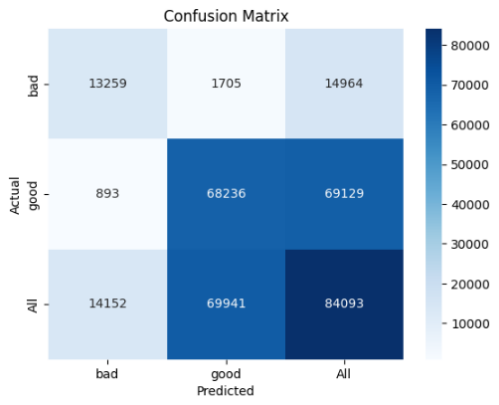

Figure 2: Confusion Matrix

Conclusion

This study presents an in-depth analysis of using machine learning for detecting malicious URLs. The model combines various features, including URL structure, domain attributes, and content characteristics, achieving an overall accuracy of 96.9%. It demonstrates particularly high performance in identifying benign URLs with a 99% accuracy. The classification report emphasizes the need to take both precision and recall into account when assessing machine learning models.

The findings highlight the model's effectiveness in phishing detection and its potential for broader cybersecurity applications. Future enhancements should focus on integrating additional features and techniques, such as deep learning and rule-based systems, to improve model performance and adaptability. The study contributes valuable insights into developing effective solutions for online threat detection.

Future Scope

1. Feature and Technique Enhancement: Future research could explore advanced deep learning architectures like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to analyze URL semantics and content.
2. Integration of Rule-Based Systems: Combining rule-based systems or blacklists with the model can reduce false positives and increase robustness.
3. Transfer Learning and Domain Adaptation: Utilizing transfer learning and domain adaptation methods can enhance performance on new datasets and adapt to evolving phishing tactics.
4. Real-Time Detection Systems: Developing real-time phishing detection systems incorporating the model and additional security measures can provide comprehensive protection against online threats.
5. Exploration of New Features: Investigating additional features such as behavioral characteristics, network traffic patterns, and user

interactions can further enhance model performance and understanding of phishing attacks.

References

1. Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection: A recent intelligent machine learning comparison based on models content and features. In 2014 IEEE Conference on E-Learning, E-Management and E-Services (IC3e) (pp. 66-71). IEEE. https://doi.org/10.1109/IC3e.2014.7081245
2. Basnet, R. B., Mukkamala, S., & Sung, A. H. (2008). Detection of phishing attacks: A machine learning approach. In Soft Computing Applications in Industry (pp. 373-383). Springer. https://doi.org/10.1007/978-3-540-77465-5_31
3. Bergholz, A., De Beer, J., Glahn, S., Moens, M. F., Paaß, G., & Strobel, S. (2010). New filtering approaches for phishing email. Journal of Computer Security, 18(1), 7-35. https://doi.org/10.3233/JCS-2009-0371
4. Jain, A. K., & Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches. Security and Communication Networks, 2017.
5. Marchal, S., Francois, J., State, R., & Engel, T. (2014). PhishStorm: Detecting phishing with streaming analytics. IEEE Transactions on Network and Service Management, 11(3), 458-471.
6. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25(2), 443-458. https://doi.org/10.1007/s00521-013-1514-3
7. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). An assessment of features related to phishing websites using an automated technique. International Journal of Computer Applications, 60(3), 19-26.
8. Patil, S., & Patil, S. (2020). Hybrid Model for Phishing URL Detection. International Journal of Advanced Computer Science and Applications, 11(4).
9. Verma, R., & Hossain, N. (2017). Semantic feature selection for text with application to phishing email detection. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. 101-112). ACM. https://doi.org/10.1145/3052973.3053008
10. Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina: A content-based approach to detecting phishing web sites. In Proceedings of the 16th International Conference on World Wide Web (pp. 639-648). ACM. https://doi.org/10.1145/1242572.1242659