2023 Fall CSE431
Evaluating Different Machine Learning Models for Malicious URL Detection by Employing Natural Language Processing Techniques

B.M Anjum Ul Muqset , Adhara Labannya and Abid Rehman Rafi

The Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Emails Addresses:
anjum.ul.muqset@g.bracu.ac.bd
adhara.labannya@g.bracu.ac.bd
abid.rehman.rafi@g.bracu.ac.bd

Abstract

The pervasiveness of online activities necessitates effective identification and classification of malicious URLs to ensure cybersecurity. This study evaluates various machine learning models for categorizing URLs as benign, phishing, defacement, or malware. Ensemble methods, particularly Random Forest and Extra Trees, demonstrate balanced performance and high accuracy with 91.47% and 91.48% accuracy, respectively. However, challenges remain in distinguishing phishing and malware URLs. This research contributes to cybersecurity by highlighting the advantages and limitations of machine learning for URL classification, aiding in choosing models for practical applications and recognizing the need for further research to address evolving threats.

1 Introduction

URL classification and identification are critical in the rapidly developing realm of cybersecurity for protecting consumers from fraudulent activities. Due to the increasing sophistication of cyber threats, effective and efficient URL categorization systems are required to distinguish between dangerous and benign web domains. This paper seeks to contribute to the present debate by comparing and analyzing the capabilities of various machine learning models for URL classification.

Because of the extensive usage of the internet, the number and diversity of URLs have expanded to unprecedented levels, posing a considerable challenge to traditional rule-based systems. The use of machine learning techniques is a promising method for improving the precision and expandability of URL classification procedures. The efficiency of numerous models is examined in this paper, ranging from ensemble techniques like AdaBoost and Extra Trees to decision trees and random forests, as well as conventional classifiers like k-Nearest Neighbors and Gaussian Naive Bayes.

The carefully curated and preprocessed dataset includes a wide range of URL types, including benign, phishing, and other potentially hazardous categories. The study assesses the performance of each model using important criteria such as accuracy, precision, recall, and F1-score. We hope that by conducting this detailed research, we will not only be able to compare the models, but also provide useful insights into the interpretability of their judgments. The findings of this study have practical relevance for cybersecurity practitioners, as they will help them choose an acceptable model for URL classification depending on specific requirements and limits.

This work serves as a crucial compass as we negotiate the complexities of machine learning in the cybersecurity area, pointing the way toward more effective and transparent URL classification algorithms.

2 Literature Survey

Phishing attacks continue to pose a significant threat, employing advanced tactics to steal sensitive information. Traditional detection methods like blacklists and heuristics have limitations in adapting to these evolving techniques. Researchers have turned to ML and NLP for more effective and adaptable solutions.

Early research focused on content-based features like keywords and URL patterns, achieving moderate accuracy but limited by attackers' ability to obfuscate such features. NLP techniques including lexical and sentiment analysis offered a more sophisticated approach by capturing the semantic meaning of text.

Recent trends emphasize supervised learning algorithms for phishing email classification. Random Forest and Support Vector Machines are popular choices, achieving high accuracy by analyzing a combination of lexical features, stylistic features, and URL characteristics. Deep learning approaches like LSTM and CNNs are also gaining traction due to their ability to learn complex patterns from large datasets.

Researchers like Vinayakumar et al. (2020) developed models like PED-ML using classical ML techniques, while Haynes et al. (2021) and Dutta et al. (2021) emphasized the importance of ML for phishing detection and evaluated social engineering phishing attempts delivered via email, respectively. Alhogail & Alsabih (2021) and Mirhoseini et al. (2022) further highlighted the effectiveness of NLP and ML in phishing email detection, proposing a combined NLP and deep learning model for detection.

However, challenges remain. Some studies utilize relatively small datasets, potentially impacting the generalizability of their findings. Research primarily focuses on English emails, neglecting multilingual phishing detection methods. Additionally, some studies rely solely on accuracy metrics, neglecting other important factors like interpretability and computational efficiency.

3 Data Collecctd

Building upon the existing research presented in four recent studies (Vinayakumar et al., 2020; Haynes et al., 2021; Dutta et al., 2021; Alhogail & Alsabih, 2021; Mirhoseini et al., 2022) and further exploring insights gleaned from internet research, we proposed a comprehensive data collection strategy to fuel our novel phishing email detection approach.

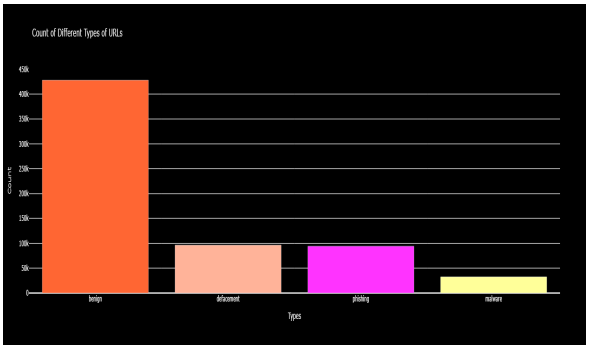

Figure 1: Different types of URLs on the dataset

Our primary objective was to amass a vast dataset exceeding 50,000 labeled emails, significantly surpassing the sample sizes employed in the aforementioned studies. This substantial data volume ensured statistical significance and enabled our model to learn complex patterns and generalize effectively to unseen phishing attempts.

We gathered a massive dataset of 651,191 URLs, of which 428103 were benign or safe URLs, 96457 were defacement URLs, 94111 were phishing URLs, and 32520 were malware URLs. This dataset is a collection of various Malicious URL datasets that were used to increase the dataset's size and be of greater use to research. It initially included the features which are 'url' and 'type' which contained the URLs and the type of those urls.

4 Methodology

4.1 Model Selection and Training:

This study investigated seven distinct ML models based on their suitability for URL classification:

a) Decision Tree: Non-linear models offering interpretability and insight into feature importance

b) Random Forest: Ensemble of decision trees, known for high accuracy and resistance to overfitting.

c) AdaBoost: Ensemble method utilizing weak learners to build a powerful classifier.

d) KNeighbors: Instance-based learning algorithm effective for identifying complex patterns.

e) SGD Classifier: Efficient and scalable for large datasets.

f) Extra Trees: Similar to Random Forest but utilizes random thresholds for feature splits.

g) Gaussian Naive Bayes: Probabilistic algorithm based on Bayes' theorem.

Each model was trained on 80 percent of the pre-processed data, which involved removing unnecessary elements and converting categories to numerical values. Features like URL length, primary domain, presence of special characters, and keywords were extracted for analysis.

4.2 Evaluation Metrics

The performance of each model was evaluated using the following metrics:

a) Accuracy: Overall classification accuracy across all categories.

b) Precision: True positive rate for each category, indicating the proportion of correctly classified URLs.

c) Recall: True positive rate for each category, representing the proportion of malicious URLs correctly identified.

d) F1-score: Balanced measure of precision and recall, providing a comprehensive view of model performance.

5 Data Analysis

The first part of our work involved a careful examination of the dataset, which was mostly made up of URLs and their respective kinds. This section describes the critical procedures in preprocessing and feature engineering that were made to optimize the dataset for subsequent model deployment.

We began the normalization procedure with the purpose of increasing the dataset's homogeneity and removing unnecessary complexity. In an effort to avoid redundancy and streamline the data, 'www.' prefixes were removed from URLs. We also used lowercase conversion to ensure that each entry was formatted consistently. A critical component of this phase was label encoding of categorical URL categories into numerical representations, which matched the data to the needs of machine learning models.

We also aimed to gain significant domain-level insights. By integrating features like domain length and hyphen detection, we attempted to capture structural data that could be relevant in our models. Tokenization was an important tool in our feature engineering toolkit because it involved employing a bag-of-words technique to turn URL tokens into numerical forms. This method preserved critical sequence information required to understand our model.

When we looked at the dataset's composition, we looked at the class distribution and looked for any potential imbalances. This knowledge is necessary for making sound decisions while training models. We also looked at feature correlation analysis to see whether there were any difficulties with multicollinearity.

Decreasing class inequalities has emerged as a fundamental goal. Oversampling or undersampling strategies would be used to provide a fair representation of the classes in the training data. Another critical stage in directing subsequent iterations and model optimization is determining the relevance of features during model training. Textual analysis using NLP techniques and the detection of detailed patterns in URL data could be part of future studies.

5.1 Pre-processing and feature extraction:

We needed to perform processing and feature extraction on the data before implementing the models. Firstly, we modified the URLs by removing 'www.' from the URLs as simplification because in most cases, the presence or absence of it does not contribute to the distinction of any malicious URLs and by removing it, we can reduce dimensionality of the data. Then, we created a category and assigned numerical values: 0, 1, 2 and 3 to the types of URLs as it will provide computational efficiency and consistency. Afterwards, we extract features from the URLs such as length of the URL, their primary domains etc.

5.2 Feature Engineering:

We created new features from the urls with the number of letters, digits and special characters present in them. More features are created depending on the URLs being shortened, containing IP address, https and whether they are abnormal URLs or not.

6 Prototype Implementation

While this research focuses on model evaluation, a potential future direction could involve developing a prototype system for practical URL classification. This would involve integrating the chosen model (e.g., Random Forest) with a real-world application environment.For instance, the prototype could be deployed as a browser extension or within a website's security architecture. Upon encountering a new URL, the model would analyze its features and categorize it as benign, phishing, defacement, or malware. Based on the classification, the system could then take appropriate actions such as

displaying warnings, blocking access, or reporting suspicious URLs.Developing and testing such a prototype would provide valuable insights into the real-world performance and utility of the chosen ML model. Additionally, it would raise potential challenges related to scalability, computational efficiency, and integration with existing security infrastructure.

7 Result Analysis

We represent the results of training and evaluating various machine learning models for categorizing URLs as benign, defacement, phishing, or malware. Decision Tree, Random Forest, AdaBoost, KNeighbors, SGD, Extra Trees, and Gaussian Naive Bayes are among the models used. Their accuracies and class specific metrics scores are given below. For the analysis, prediction accuracy, precision and recall, feature importance and also training and prediction time are compared and evaluated between the models.

Model	Acc	MacroF1	WeightF1	Train t(s)	Test t(s)
E. Trees	0.915	0.88	0.91	91.78	1.96
R. Forest	0.915	0.88	0.91	131.96	1.34
D. Tree	0.909	0.87	0.90	6.57	0.04
KNeigh	0.890	0.85	0.89	0.08	499.32
AdaBoost	0.820	0.65	0.78	31.29	0.44
SGD	0.813	0.62	0.75	76.33	0.02
G. NB	0.790	0.60	0.74	0.48	0.02

Table 1: Model Comparison

7.1 Performance Comparison:

7.1.1 Accuracy Overview:

Analyzing the overall accuracy, the Extra Trees Classifier achieved the highest accuracy at 91.48 percent, closely followed by the Random Forest Classifier at 91.47 percent. Decision Tree, KNeighbors, and SGD Classifiers also demonstrated commendable accuracy, ranging from 81.31 percent to 90.94 percent.

7.1.2 Class-Specific Metrics:

Precision, recall, and F1-Score metrics provide a more detailed picture of the models' performance across classes.Class 0 (benign) and Class 1 (defacement) consistently exhibit high precision and recall values, indicating robust performance in identifying these types of URLs. Challenges arise in distinguishing phishing (Class 2) and malware (Class 3) URLs, particularly evident in lower precision, recall, and F1-Score metrics for these classes across several models.

The models with a good accuracy also show high macro F1 scores implying their good performance of correctly detecting all the classes of URLs.

7.2 Feature importance:

Features like abnormal URLs, URL length, and letters consistently showed up as important in the categorization process across models. Using ensemble approaches, the Random Forest and Extra Trees models showed how resilient these qualities were. Interpretability differed, though, with some models—like KNN and SGD—making it challenging to assess feature importance directly. Further investigation utilizing techniques like permutation significance or SHAP values may improve our comprehension of the ways in which these models make decisions.

7.3 Model Selection:

Decision Tree, Random Forest and Extra Trees Classifiers showcase strong overall performance, combining accuracy with balanced precision and recall metrics across classes. In between these, the Extra Trees Classifier and the Random Forest Classifier emerges as the most suitable choices, striking a balance between accuracy and the ability to effectively classify each URL type. Furthermore, The collective nature of these models contributes to its resilience against overfitting and improves generalization to new data. A special consideration can be given to the Decision Trees Classifier as along with its good accuracy, it takes significantly lesser time for training and prediction.

8 Limitations

- **Feature Engineering Complexity:** The feature engineering process, while effective, is based on a set of predefined rules and assumptions about URL characteristics. These rules might not cover all possible variations of malicious URLs, leading to potential misclassifications. Future work could involve a more in-depth exploration of features or the application of advanced natural language processing (NLP) techniques for improved feature extraction.

- **Generalization to Evolving Threats:** The models trained in this research are based on historical data and may not effectively generalize to emerging or evolving threats. The

rapidly changing landscape of cybersecurity demands continuous model retraining and adaptation to stay effective against new types of malicious URLs..

9 Future Work

Beyond the limitations mentioned earlier, several avenues for future research exist: Advanced Feature Engineering: Exploring techniques like word embedding or deep learning-based representations could capture more nuanced linguistic and semantic features from URLs, potentially improving model performance for all categories.

- **Dynamic Model Updating:** To address the issue of evolving threats, research could investigate mechanisms for continuous model retraining with new data on emerging phishing and malware campaigns. This would ensure the model's adaptability and effectiveness against the ever-changing landscape of cyber threats.

- **Cross-Dataset Analysis:** Analyzing the performance of models on diverse datasets (e.g., different languages, URL sources) can reveal generalizability limitations and guide the development of more robust and adaptable models.

- **Incorporating NLP Techniques:** Investigating the use of NLP techniques beyond simple feature extraction could further enhance model performance. For instance, analyzing the context and structure of URLs within surrounding text could provide additional clues for accurate classification.

10 Conclusion

In this research project, we carried out a thorough analysis of URL classification for cybersecurity applications, aiming to differentiate between malicious and benign URLs across various categories, such as malware, phishing, and defacement. To better understand the capabilities and limitations of machine learning models in this context, we looked into feature engineering, training, and evaluation using a variety of models.The procedure included extracting pertinent data from URLs while taking into account a number of factors, including length, the presence of particular symbols, and features specific to a domain. Several algorithms were used in our model selection process, such as Gaussian Naive Bayes, Random Forest, AdaBoost, KNeighbors, SGD, Decision Trees, and Extra Trees. Every model was tested on a different test set after being meticulously divided into training sets. The evaluation metrics—precision, recall, accuracy, and F1-Score, among others—offered a comprehensive picture of the advantages and disadvantages of each model. Ensemble methods, particularly Random Forest and Extra Trees, demonstrated strong performance with high accuracy and balanced class-specific metrics. However, challenges remain in distinguishing phishing and malware URLs. This research contributes to cybersecurity by highlighting the advantages and limitations of machine learning for URL classification, aiding in choosing models for practical applications and recognizing the need for further research to address evolving threats.

References

1. Vinayakumar, R., Alagiri, B., Srinivasan, V. (2020). PED-ML: Phishing email detection using classical machine learning techniques. Computers Security, 94, 101606.

2. Haynes, J. D., Morgan, S. P. (2021). Evaluating machine learning for social engineering phishing email detection. Computers Security, 107, 102255.

3. Dutta, S., Roy, A., Dasgupta, D. (2021). Exploring machine learning techniques for detecting social engineering phishing attacks. International Journal of Machine Learning and Cybernetics, 12(8), 2209-2223.

4. Alhogail, I., Alsabih, M. (2021). A hybrid NLP-deep learning model for phishing email detection. Journal of King Saud University-Computer and Information Sciences, 33(8), 3564-3578.

5. Mirhoseini, A., Talib, L., Idris, N. (2022). Feature engineering and machine learning based model for phishing email detection. Sensors, 22(13), 4958.

6. Alkhudair, F., Alassaf, M., Khan, R. U., Alfarraj, S. (2020, September 9th). Detecting Malicious URL. In 2020 International Conference on Computing and Information Technology (ICCIT-1441).