Malicious URL Detection

Adnan Azem, Jode Shibli

February 2024

Abstract:

Phishing presents a significant challenge distinct from other security risks like intrusions and malware, which exploit technical vulnerabilities in network systems. The vulnerability of any network lies in its users. Phishing URLs primarily aim at individuals and organizations through social engineering tactics, exploiting human weaknesses in information security awareness. These URLs entice online users to visit fraudulent websites, where their confidential data, including debit/credit card details and other sensitive information, are harvested.

1 *Introduction*

As information technology continues to advance rapidly, we find ourselves increasingly vulnerable to cybercrime. The Internet has evolved into a fundamental component of modern life and a crucial driver of technological progress, enabling efficiencies in time, effort, and costs.

This study centers on a social engineering-driven URL phishing attack aimed at individuals. It involves the creation of deceptive websites designed to trick victims into divulging sensitive information, such as email credentials, credit card details, and other confidential data, thereby potentially tarnishing the reputation of individuals or institutions.

Numerous recent studies have sought effective solutions for detecting phishing URLs, which can be categorized into four main classifications: predefined lists, signature-based methods, content-based approaches, and machine learning techniques.

This study will concentrate on the machine learning classification method. This approach relies heavily on learning the features of websites categorized as phishing, then applying predictive capabilities to differentiate between genuine and fake websites using various machine learning techniques such as prediction and classification.

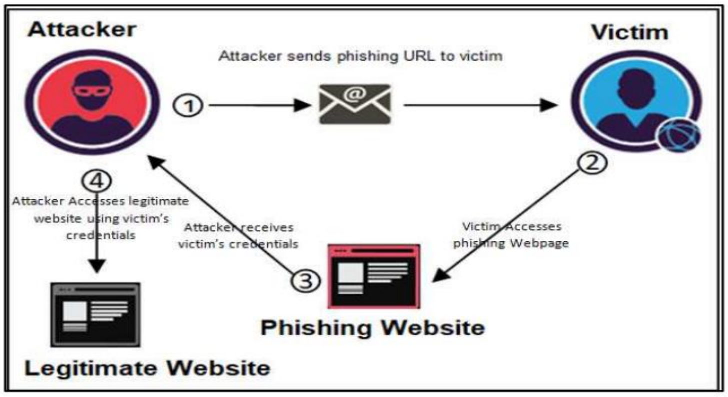
Attacker
Victim
Attacker sends phishing URL to victim
1
2
3
4
Attacker Accesses legitimate website using victim's credentials
Attacker receives victim's credentials
Victim Accesses phishing Webpage
Phishing Website
Legitimate Website

Fig 1. Website Phishing Lifecycle

2 *Related Work*

Nowadays, many anti-phishing techniques are proposed, but still, there is a challenge to get high accuracy detection with a low ratio of false-positive detection. In this section, a review of related work techniques and their features is presented.

The approach proposed in [1] is a real-time detection system using URL features only; a dataset of 46,5461 URLs was used with three classifiers (J48, SVM, and Logistic Regression), which were implemented using WEKA software; the highest accuracy was 93% which was gained by J48 classifier.

Authors et al. [2] implemented a middleware system to detect phishing websites. Multiple algorithms, including Random Forest, SVM, and K-Nearest Neighbor (KNN); a dataset of 11055 URLs were collected from UCI and narrowed down to contain 22 features, the highest accuracy (96%) was obtained using RF algorithm.

Another model proposed by the authors in [3] using a URL identification strategy utilizing the Random Forest algorithm. A dataset was gathered from PISHTANK; only 8 out of 30 features were used for analysis. Finally, an accuracy of 95% was achieved by this model. Where authors in [4] proposed a system PHISH-SAFE using SVM and Naïve Bayes (NB) classifiers; the results show the highest accuracy 90% with the SVM.

Authors et al. [5] used random forest algorithm and compared the result with (Logistic Regression, J48, and Naïve Bayes) algorithms. Random Forest algorithm gained the best result with accuracy of (86.9%). Where authors in [6] proposed a new design called Extreme Learning Machine (ELM) based on the RF algorithm using 30 URL features; ELM detecting accuracy was 95.34%.

From another perspective, authors in [7] proposed a technique through content analysis and URL features extraction. Artificial Neural Network, Decision Tree, Support Vector Machine, K-Nearest Neighbor and Naive Bayes algorithms were used in this approach. The highest accuracy (96.01%) was obtained using Artificial Neural Network algorithm.

Table 1 Summarization of previous works and their techniques

Author	Used algorithm	Accuracy
[1]	J48, SVM and LR	93% using J48
[2]	RF, SVM and kNN	96% using RF
[3]	RF	95%
[4]	SVM and NB	90% using SVM
[5]	RF, J48, NB and LR	86.9% using RF
[6]	RF	95.34%
[7]	DT, ANN, NB, SVM and kNN	96% using ANN

3 Methodology

Fig. 2 describes the high-level methodology and the following steps in this work. Also, for each step in the figure, the following subsections explain the details.

3.1 *Dataset Description*

In this work we had two different datasets:

The first dataset we found at Kaggle website which is balanced dataset, because the creation of the dataset has involved 2 different datasets from Kaggle which are as follows:
First Dataset: 450,176 URLs, out of which 77% benign and 23% malicious URLs.
Second Dataset: 651,191 URLs, out of which 428103 benign or safe URLs, 96457 defacement URLs, 94111 phishing URLs, and 32520 malware URLs.
To create the **Balanced dataset**, the first dataset was the main dataset, and then more malicious URLs from the second dataset were added, after that the extra Benign URLs were removed to keep the balance. Of course, unifying the columns and removing the duplicates were done to only keep the unique instances.

The second dataset we found at Kaggle website which has been collected from Alexa website ranking a blacklist of previous DGA domain names.
This dataset contains 4 files two DGA files, one alexa ranking dataset and an english words dataset.
The first file contains the name of the DGA (irrelevant IMO for just building a classifier), domain (most importanta information) name and a timestamp.
What is important to keep as information are the domain names, the rest can be dropped as we will do some feature engineering to create relevant columns.
The second file contains legitimate domain names from Alexa, domain names are ranked by their popularity.
The third file is like the previous but a bit longer.
The fourth file is a dictionary of English words collected from GitHub. This file will be used to compare ngrams from domain names.

In each of the datasets we split the data in an 80:20 ratio, 80% used for training and the remaining 20% is used for testing.

3.2 Feature Extraction

Like we stated above in the dataset description section, we had 2 different datasets. So for each data set we trained a number of classification models, with different features for each dataset.

In the first dataset, we extracted 19 features overall and ended up using 17 of them. We categorized them into 3 groups: 1, Length based features. 2, Count based features. 3, Binary features.

The features in the first group are: Length of URL, Length of Host-name, Length Of Path, Length Of First Directory, Length Of Top Level Domain.

The features in the second group are: Counts of ’@’, ’?’, ’%’, ’.’, ’=’, ’http’, ’https’, ’www’. As well as number of digits, letters and re-directions.

The third group contained: if the URL is using and IP address and if the URL is using a shortening service.

In the second dataset, we first extracted 18 overall and ended up using all of them. We categorized them into 3 groups: 1, Structural Features. 2, Linguistic Features. 3, Statistical Features.

The features in the first group are: Domain Name Length, Number of Subdomains, Subdomain Length Mean, Has www Prefix, Has valid TLD, Contains Single-Character Subdomain, Contains TLD as Subdomain, Underscore Ratio, Contains IP Address

The features in the second group are: Contains Digits, Vowel Ratio, Digit Ratio, Ratio of Repeated Characters, Ratio of Consecutive Consonants, Ratio of Consecutive Digits.

The third group contained: Entropy, words gram, alexia gram.

3.3 Dataset Training and Testing Splits

Both datasets were prepared for the training and testing using adopted machine-learning approach, where 80% of its instances were used for training, and the remaining 20% for testing.

3.4 Experiments

For the first Dataset we trained 4 models:

I. ***Logistic Regression***
The most used statistical model for predicting binary data in various disciplines is logistic regression.

II. ***Random Forest***
Random Forests, another ensemble-learning algorithm, was adopted to conduct the binary classification process on the URL dataset at hand.

III. ***Decision Tree***
An improved version of classification and regression trees is the decision tree algorithm.

IV. ***XGBoost***
XGBoost (short for **Extreme Gradient Boosting**) is a powerful machine learning algorithm known for its efficiency, speed, and accuracy. It belongs to the family of boosting algorithms, which are ensemble learning techniques that combine the predictions of multiple weak learners.

For the second Dataset we trained one model:

1. ***Lightbgm***
   LightGBM (short for **Light Gradient Boosting Machine**) is a powerful gradient-boosting framework used in machine learning, it's based on decision tree algorithms and finds applications in ranking, classification, and other machine learning tasks.

2. ***Gradient Boosted***
   Gradient boosting classifier is a set of machine learning algorithms that include several weaker models to combine them into a strong big one with highly predictive output. Decision trees are usually used when doing gradient boosting.

3. ***Random Forest***

4. ***XGBoost***

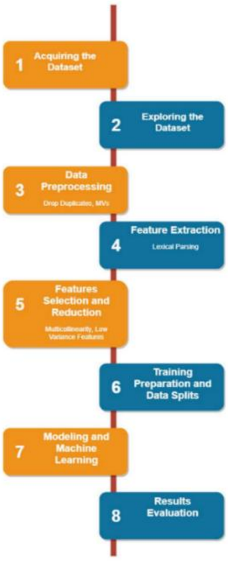
1 Acquiring the Dataset
2 Exploring the Dataset
3 Data Preprocessing
Drop Duplicates, MVs
4 Feature Extraction
Lexical Parsing
5 Features Selection and Reduction
Multicollinearity, Low Variance Features
6 Training Preparation and Data Splits
7 Modeling and Machine Learning
8 Results Evaluation

Fig 2. Work Methodology

4 *Results*

From the first dataset with the 4 models that we trained them we obtained the following accuracy and f1:

1. From Logistic Regression we obtained: 99.49% accuracy and 99% F1-score.
2. From Random Forest we obtained: 99.64% accuracy and 100% F1-score.
3. From Decision Tree we obtained: 99.45% accuracy and 99% F1-score.
4. From XGBoost we obtained: 99.64% accuracy and 100% F1-score.

From the second dataset with the 4 models that we trained them we obtained the following accuracy:

1. From Lightbgm we obtained: 98.54% accuracy.
2. From Random Forest we obtained: 98.71% accuracy.
3. From XGBoost we obtained: 98.68% accuracy.
4. From Gradient-boosted we obtained: 97.64% accuracy.

5 *Summary*

Our research revealed that accurately identifying random phishing URLs with 99% precision was straightforward, drawing on our findings and earlier research. Nonetheless, the ongoing emergence of fresh phishing sites remains an ongoing issue. In response, we gathered extra data produced by Domain Generation Algorithms (DGAs) and developed a novel model to complement the pre-existing ones, thereby broadening the spectrum of identifiable phishing sites. Our findings indicate that employing a combined strategy featuring the top-performing models from both data sets enhances the probability of successfully detecting phishing domains.

6 *References*

1. URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis.
2. Phishing URL detection using machine learning methods.
3. FANCI: Feature-based Automated NXDomain Classification and Intelligence.