Cybersecurity: Phishing Website Classification

Devananth V - EP20BTECH11004

***Abstract*— Phishing has been recognized as the easiest and the most widespread cybersecurity threat as it targets the weakest link in the security chain, the User. This paper is dedicated to the identification of phishing websites, employing Machine Learning Techniques (MLTs) as a robust mechanism. The dataset utilized for this research is sourced from 'Mendeley data,' comprising 11,430 URLs with 87 extracted features. The dataset is balanced, it contains exactly 50% phishing and 50% legitimate URLs. Our primary objective is to conduct a comparative analysis of various machine learning models, aiming to identify the optimal model through a process of systematic optimization. This research contributes to the advancement of cybersecurity by leveraging empirical data and rigorous model evaluation to enhance the precision and efficacy of phishing website identification.**

I. Introduction

Phishing is the practice of sending fraudulent communications that appear to come from a reputable source. The goal is to steal sensitive data like credit card and login information or install malware on the victim's machine. Hackers do not need to crack any complex cipher code neither breach a hard firewall. Instead, they simply send emotional, critical or sensible e-mails urging recipients to introduce their personal credentials by clicking on a link. Recipients are then redirected to fake web that look very similar to those targeted authentic websites.

During the pandemic, there was a notable surge in phishing activities. This increase can be attributed to the heightened demand for current information during this period, which presented an opportunity for malicious actors to impersonate trusted entities, such as health services and government websites.

To get a better understanding let us consider a case in our own IIT Hyderabad campus:

A simulation attack was conducted by the Computer Center, IIT Hyderabad in collaboration with Kludge, networking club IIT Hyderabad as part of their Cybersecurity campaign. Their mode of attack was phishing emails. The subjects of the above mentioned phishing emails were:

- WiFi Password Expiration Notification.
- Avail the Chat GPT Plus Subscription.
- Reviewing the APAR Appraisal Form.
- Additional Funds Allocated to Your Department.
- Urgent Report Submission.

These emails were sent from the domain "iith-ac.c-0m.com".

The use of special characters in the above domain and its resemblance to an existing domain is an indication for it to be a phishing website. We choose about 87 features like this from its URL, content and information from external records to figure out its legitimacy.

Thus Machine learning is useful for classifying phishing websites because it offers adaptability to evolving attack methods, enables automated and large-scale analysis of datasets, excels in pattern recognition, provides real-time detection, reduces false positives, enhances accuracy through continuous learning, and allows for behavioral analysis, all of which contribute to effective and efficient identification of phishing threats.

II. Literature Review

"TOWARDS BENCHMARK DATASETS FOR MACHINE LEARNING BASED WEBSITE PHISHING DETECTION: AN EXPERIMENTAL STUDY"

Link: 'https://arxiv.org/pdf/2010.12847.pdf'

A. Objective

To propose and experiment a general scheme for building reproducible and extensible datasets for website phishing detection, and to evaluate the performance of different machine learning techniques and features on the built dataset.

B. Methodology

URLs are collected from various sources and preprocessed, from which 87 features are extracted under different classes(URL, Content, External). The dataset is then balanced and shuffled. Two metrics are used for performance

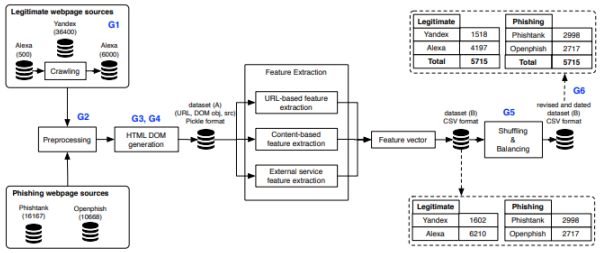

Fig. 1: Dataset Preparation

Evaluation, Accuracy and Macro F1 Score.

The classifiers used are:

- Naive Bayes
- Support Vector Model
- Logistic Regression
- Decision Tree
- Random Forest

The following experiments are the done:

- Best model for single or combination of feature classes
- Combination of different models
- Feature selection
- Feature extraction runtime analysis

C. Results

A scheme for constructing benchmark datasets that can enable comparison, replication, and extension of website phishing detection systems was presented.

Random Forest was found to give the best accuracy for all the classes of features. External-based features provide the best accuracy, followed by URL-based and then Content-based. Using hybrid features was found to be better than using single class of features.

It was found that chi-square filter gave the best accuracy with less number of features and that filter ranking together with incremental removal of less important features improved the performance. Wrapper methods failed to improve the accuracy compared with filter methods and using 'Boruta algorithm' was much faster than 'WrapperSubsetEval'.

Extraction time of different features and feature classes was investigated. It was found that content-based features were the most time consuming and that some hyperlink-based features also caused severe network delays.

III. Methodology

A. Dataset Description

The Dataset at 'Mendeley data' was designed to be used as a a benchmark for machine learning based phishing detection systems. The dataset includes 11430 URLs with 87 extracted features. The dataset is balanced, it contains exactly 50% phishing and 50% legitimate URLs. The Datasets was constructed as of May 2020.

Link: 'https://data.mendeley.com/datasets/c2gw7fy2j4/3'

It has features from three different classes: 56 extracted from the structure and syntax of URLs, 24 extracted from the content of their correspondent pages and 7 are extracted by querying external services.

The datasets are divided into 2:

- Dataset A: contains a list a URLs together with their Document Object Model (DOM) tree objects.
- Dataset B: contains the extracted feature values that can be used directly as input to classifiers for examination.

B. Workflow

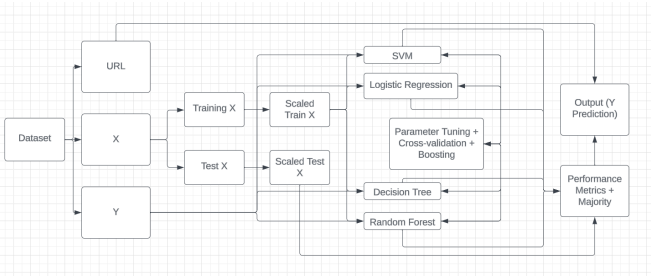

Fig. 2: Workflow

1) Extracting the data: We get the 'x' value by dropping the columns 'URL' and 'status' from the dataset. 'y' value from the status column of the dataset by assigning '1' for legitimate and '0' for phishing. We split the data into training and testing/validation data, with 30% of the data as testing data.

Test Data Size	Training Data Size
9140	2285

TABLE I: Test-Train Split

2) Scaling the data: Standardization is used to scale the 'X' values, centering it around the mean with a unit standard deviation. We use the 'fit_trasfrom' and 'transform' function within 'StandardScalar' class to standardize the training data and test data respectively.

3) Models Selection: We consider the four models:

- Support Vector Classifier.
- Logistic Regression Classifier.
- Decision Tree Classifier.
- Random Forest Classifier.

4) Parameter Tuning + Cross Validation: GridSearchCV is a technique in machine learning that systematically explores a predefined hyperparameter grid to find the optimal configuration for a model. It utilizes 'k-fold cross-validation' to evaluate performance for each hyperparameter combination and selects the best parameters based on the average performance.

5) Boosting(Adaboost): Boosting is an ensemble method, which means it combines several "weak" models to create a "strong" model. The weak model is used to make predictions on the dataset, and the misclassified instances are identified. A new model is then trained, giving more importance or weight to the instances that were misclassified by the previous model. This process is repeated several times, with each new model focusing on the errors of the previous model. All the models are combined to make a final prediction.

6) Model Comparison: We use the following metrics to compare the models:

- Time Taken
- Accuracy
- Macro F1 Score
- Recall
- Log Loss

7) Output: To increase the accuracy of the prediction we take the majority decision given by the four models. We output 'legitimate' if the majority decision is of class '1' and in the case of class '0', the output is given as 'phishing'.

C. Support Vector Model

It is a part of Supervised Learning paradigm.

The objective function has two parts: the first aims to maximize the separation between classes, while the second works to minimize misclassifications. The parameter "C" is the cost assigned to mistakes, treating errors in both positive and negative cases equally.

1) Objective Function:

$$\min(\|w\|^2 + C\sum_{i=1}^{n}\xi_i) \quad \text{where } y_i(w\cdot f(x_i)+b) \geq 1-\xi_i \quad \text{and } \xi_i \geq 0 \quad (1)$$

2) GridsearchCV Parameters: We tune the parameter 'C' using the values '0.01, 1, 10, 100'. We use 10 fold cross-validation.

D. Logistic Regression

It is a part of Supervised Learning paradigm.

The algorithm applies the logistic function (sigmoid) to a linear combination of input features, producing a probability score. A threshold is then set to classify instances into one of the two classes.

1) Objective Function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \quad (2)$$

$h_\theta(x)$ is the sigmoid (logistic) function applied to the linear combination of input features x.

2) GridsearchCV Parameters: We tune the parameters 'C' using the values '0 , 0.25, 0.5 , 0.75, 1'. We use 10 fold cross-validation.

E. Decision Tree

It is a part of Supervised Learning paradigm.

It works by recursively partitioning the data into subsets based on the most significant attribute at each node. The goal is to create a tree-like structure where each internal node represents a decision based on a feature, each branch represents the outcome of that decision, and each leaf node represents the final prediction or value.

1) Objective Function: Decision tree does not have a single objective function that summarizes the entire tree-building process, but you can express the impurity or information gain at a specific node using a criterion such as Gini impurity or entropy.

$$\text{Gini index } G = \sum_c \Pi_c(1-\Pi_c)$$

where

$$\Pi_c = \frac{\text{no.of points in the subspace with label } c}{\text{no.of points in the subspace}}$$

2) GridsearchCV Parameters: We tune the parameters max_depth with the values '1, 2, 4, 6, 8, 10, 12'. We use 10 fold cross-validation.

Metric	SVM	Logistic Regression	Decision Tree	Random Forest
Time Taken	18.614	20.642	7.521	34.35
Accuracy	0.969	0.961	0.957	0.976
Precision	0.97	0.962	0.959	0.978
F1-score	0.965	0.954	0.943	0.975
Log Loss	0.088	0.033	0.049	0.044
Recall	0.440	0.529	0.265	0.412

TABLE II: Model Comparison

F. Random Forest

It is a part of Supervised Learning paradigm.

Random Forest is an ensemble learning method that combines the predictions of multiple decision trees to improve overall performance and robustness. It operates by constructing a multitude of decision trees during training and outputs the mode (classification) or average (regression) prediction of the individual trees. Random Forests introduce randomness in two key ways: by using a random subset of the training data for each tree (bootstrapping) and by considering a random subset of features at each split. This helps reduce overfitting and increases the model's generalization ability.

1) Objective Function: As it is a ensemble of decision trees there isn't a single objective function that summarizes the entire Random Forest; instead, the focus is on the individual decision trees and their combination.

2) GridsearchCV Parameters: We tune the $n_estimators$ parameter with the values . We use 10 fold cross-validation.

IV. Experimental Results

In our pursuit to develop an effective phishing website classifier, we successfully identified a robust model. Through the process of parameter tuning, we fine-tuned the various settings of the classifier, enhancing its accuracy and ensuring optimal performance. This allowed us to achieve a level of precision that meets the high standards required for accurately distinguishing phishing websites from legitimate ones. Employing a 'majority voting' approach among the models enhances accuracy and yields satisfactory results.

Recognizing the importance of guarding against overfitting, we implemented k-fold cross-validation. This technique involves partitioning our dataset into k subsets, training the classifier on k-1 of these folds, and validating its performance on the remaining fold. Repeating this process k times with different validation sets allows us to comprehensively assess the model's generalization capability. By preventing overfitting, we ensure that our classifier doesn't become excessively tailored to the training data, maintaining its ability to make accurate predictions on unseen data.

The successful combination of parameter tuning and k-fold cross-validation not only led to a well-performing classifier but also instills confidence in its reliability across different datasets. The confusion matrix of each of the machine learning models is given with the table comparing their performance metrics.

	URL	Actual	Predicted
1	http://www.crestonwood.com/router.php	legitimate	legitimate
2	http://appleid.apple.com-app.es/	phishing	phishing
3	http://http://html.house/l7ceeid6.html	phishing	phishing
4	http://rgipt.ac.in	legitimate	legitimate

TABLE III: Example Results

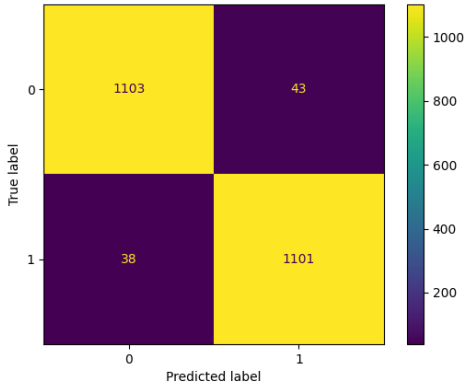

Fig. 3: Support Vector Classifier

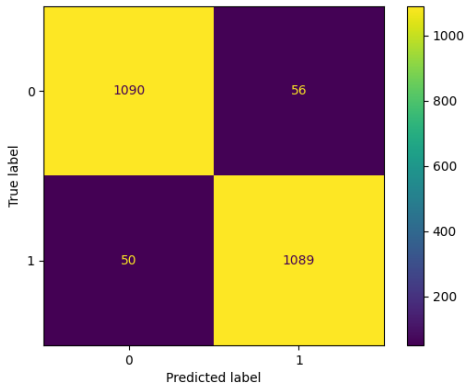

Fig. 4: Logistic Regression Classifier

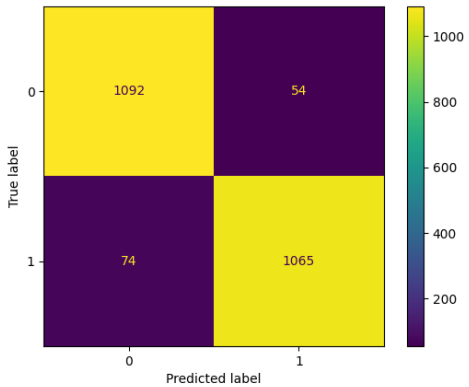

Fig. 5: Decision Tree Classifier

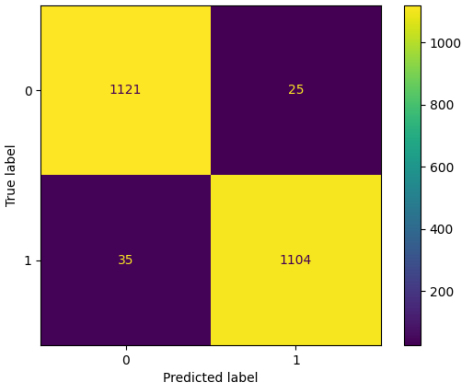

Fig. 6: Random Forest Classifier

V. Discussion

- From the confusion matrix it is evident that the Random Forest classifier gives the least False Negative and False Positives.
- The Random forest classifier also gives the best accuracy and macro-f1 score.
- Support Vector takes an average time as well as gives good accuracy.
- Prioritizing time we find Decision tree classifier takes less time with a small loss in accuracy.
- Logistic Regression takes longer than svm but gives a lower accuracy.

VI. Improvements

- In this case we used all the available features as parameters, but their combination can also be tried to check if better results can be obtained.
- We could also try to increase performance by selecting featured ranked using filter methods like 'chi-square', 'Pearson co-relation', etc.
- We could also try making it a real-time classification, like as in the case of a extension in a websites. However, in case the run-time for feature extraction should be decreased.

References

- New DNS Threat Trend Report - Cisco Umbrella
- Web page phishing detection - Mendeley Data
- 2010.12847.pdf (arxiv.org)