DATA WAREHOUSE AND MINING REPORT

(DWDM - SE407a)

Department of Computer Science and Engineering
Delhi Technological University

Submitted to -

Dr. Sonika Dahiya

Submitted by -

Kevin Mirchandani (2K18/SE/093)
Ishaan Jain (2K18/SE/067)
Dhruv Yadav (2K18/SE/056)

ABSTRACT:

With the use of machine learning, we will look to reduce human effort in recognising, learning, predictions and many more areas. The project will be able to recognise phishing and non-phishing websites, comparing classifiers like K Nearest Neighbours and Naive Bayes Classifier on the basis of performance, accuracy, positive productivity, and specificity with using different parameters with the classifiers.

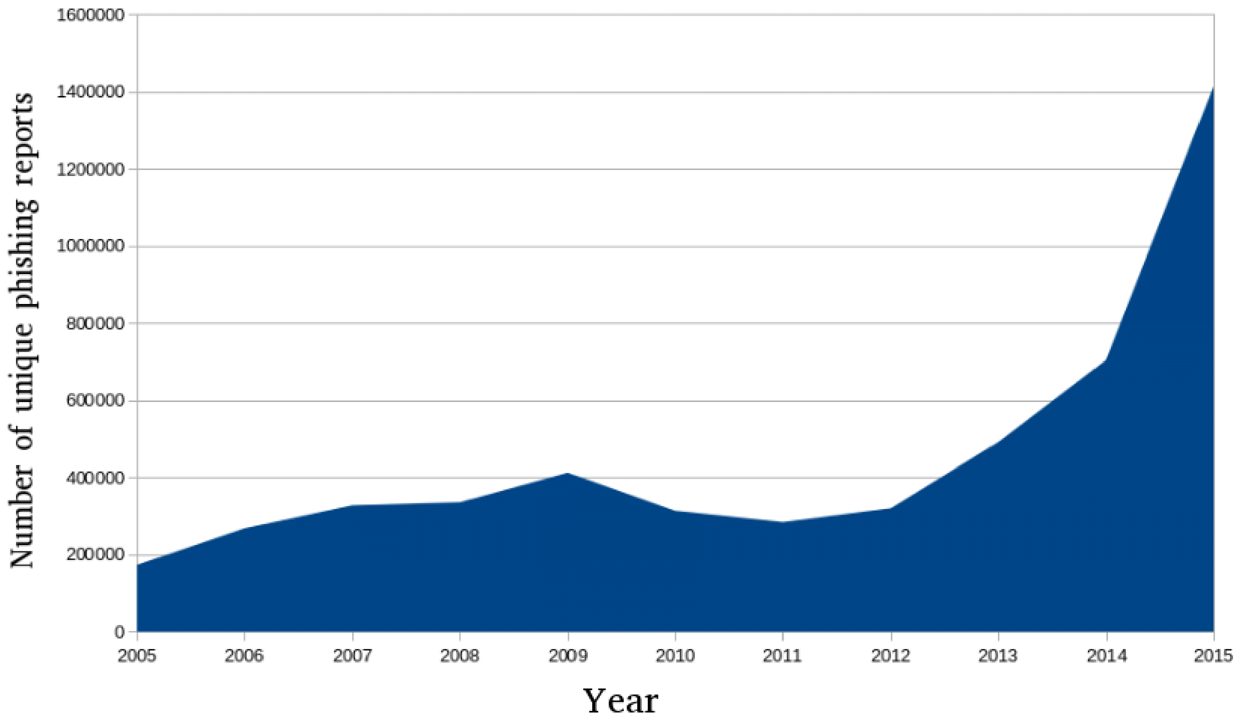
The Internet has become an indispensable part of our life, However, It also has provided opportunities to anonymously perform malicious activities like Phishing. Phishers try to deceive their victims by social engineering or creating mockup websites to steal information such as account ID, username, password from individuals and organizations. Although many methods have been proposed to detect phishing websites, Phishers have evolved their methods to escape from these detection methods. One of the most successful methods for detecting these malicious activities is Machine Learning. This is because most Phishing attacks have some common characteristics which can be identified by machine learning methods. In this paper, we compared the results of multiple machine learning methods for predicting phishing websites.

INTRODUCTION:

Phishing is a kind of Cybercrime trying to obtain important or confidential information from users which is usually carried out by creating a counterfeit website that mimics a legitimate website. Phishing attacks employ a variety of techniques such as link manipulation, filter evasion, website forgery, covert redirect, and social engineering. The most common approach is to set up a spoofing web page that imitates a legitimate website.

The COVID-19 pandemic has boosted the use of technology in every sector, resulting in shifting of activities like organising official meetings, attending classes, shopping, payments, etc. from physical to online space. This means more opportunities for phishers to carry out attacks.

Detecting phishing websites is not easy because of the use of URL obfuscation to shorten the URL, link redirections and manipulating links in such a way that it looks trustable and the list goes on. This necessitated the need to switch from traditional programming methods to machine learning approaches.



EXISTING SOLUTIONS:

Traditionally, the ad-hoc methods have been used to detect phishing attacks based on content, URL of the webpage, etc. There are primarily three modes of phishing detection:

1. **Content-Based Approach:**

   Analyses text-based content of a page using copyright, null footer links, zero links of the body HTML, links with maximum frequency domains. Using only a pure TF-IDF algorithm, 97% of phishing websites can be detected with 6% false positives.

2. **URL Based Approach:**

   Uses page rank and combines it with other metrics derived from URL based on a priori knowledge. This method can detect up to 97% of phishing websites.

3. **Machine Learning Approach:**

   Uses different machine learning models trained over features like if URL contains @, if it has double slash redirecting, pagerank of the URL, number of external links embedded on the webpage, etc. This approach could get upto 92% true positive rate and 0.4% false positive rate.

DATASET:

Dataset Description

We used the dataset provided by UCI Machine Learning repository. The dataset has 11055 data points with 6157 legitimate URLs and 4898 phishing URLs. Each datapoint had 30 features subdivided into following three categories:

1. URL and derived features
2. Page's source code based features: Includes URLs embedded in the webpage and HTML and Javascript based features.
3. Domain based features

Studying the way of extraction and relevance of features, we dropped 5 features out of 30, namely: Port Number, Abnormal URL, Pop-up Window, Google Index and Number of Links Pointing to a Page. Port Number was dropped due to feature drift. Rest were dropped due to unavailability of methods to extract them programmatically or absence of public APIs.

Visualizing the dataset

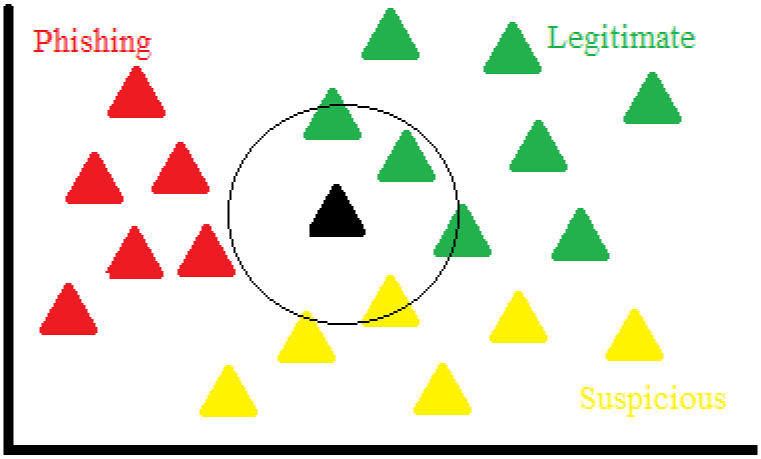
To see separability of the two classes, we plotted the t-SNE curve. The curve implied that though the classes are separable, they are not clustered together, and either transformation of the features or non-linear model is required to obtain good results.

We splitted the available data into training and testing data using 80:20 split. Post that, since we had only 7075 data points in the training data, we trained it using 5 fold cross validation. Hence, we achieved a train:val:test split of 64:16:20. We one-hot encoded the features to avoid any biases due to numerical values.

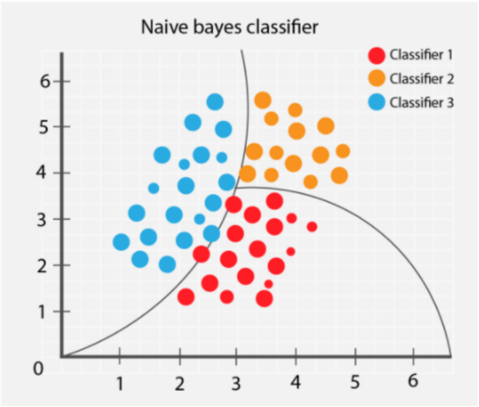
MODEL DETAILS:

We tried out various classification models with hyperparameter tuning:

1. **K-Nearest Neighbours**: KNN calculates the nearest k neighbours for each data point and returns the majority label among them. The hyperparameters used are n neighbours as 3 and the metric for distance evaluation is ‘euclidean’ distance.



2. **Categorical Naive Bayes**: It is a probabilistic model that assumes the features to be independent of each other.



EVALUATION METRICS:

For testing the results obtained, we used 3 parameters: Accuracy, Recall and False Positive Rate (FPR).

1. ***Accuracy**:*

   It is the ratio of the number of correct predictions to the total number of input samples. Since the objective requires most of the URLs to be classified correctly, hence high accuracy is one of the metrics.

2. ***Recall**:*

   It is the ratio of the number of true positives to the total number of predicted positives. As we want the websites predicted as positive, to be legitimate only, high recall is desired.

3. ***False Positive Rate (FPR)**:*

   It is the ratio of the number of samples incorrectly identified as positive to the total number of actually negative samples. The requirement is to minimise the number of phishing websites identified as legitimate as it can lead to heavy losses for the person visiting the website. Thus low FPR is one of the metrics used.

- **KNN :**

To accomplish this goal, we used our 6-step pipeline to train the model:-

Step 1: Structuring our initial dataset

Step 2: Splitting the dataset into train and test dataset

Step 3: Normalising the dataset

Step 4: Calculating Euclidean distance

Step 5: Get nearest neighbours

Step 6: Make predictions

- **NAIVE BAYES CLASSIFIER :**

To accomplish this goal, we used our 7-step pipeline to train the model:-

Step 1: Structuring our initial dataset

Step 2: Splitting the dataset into train and test dataset

Step 3: Separate by class

Step 4: Summarize the dataset

Step 5: Summarize the dataset by class

Step 6: Gaussian probability density function

Step 7: Calculate class probabilities

RESULT AND DISCUSSIONS:

KNN:

Accuracy - 93.57% with k = 5

Recall - 93.5%

False Positive Rate(FPR) - 5.44%

Naive bayes classifier:

Accuracy - 91.49%

Recall - 91%

False Positive Rate(FPR) - 8.03%

In this project, Our methodology uses not just traditional URL based or content based rules but rather employs the machine learning technique to identify not so obvious patterns and relations in the data. We were able to obtain an accuracy of **93.57%**, recall of **93.5%** with a False Positive Rate of **5.44%**, thus classifying most websites correctly and proving the effectiveness of the machine learning based technique to attack the problem of phishing websites.

By looking at the Evaluation metrics, it can be concluded that K Nearest Neighbours(KNN) performs better than Naive Bayes Classifier on this dataset.

REFERENCES:

[1]https://towardsdatascience.com/whataphish-detecting-phishing-websites-e5e1f14ef1a9

[2] Ankit Jain and B B Gupta. Phishing detection: Analysis of visual similarity based approaches. Security and Communication Networks, 2017:1–20, 01 2017.

[3] R. M. Mohammad, F. Thabtah, and L. McCluskey. UCI machine learning repository, 2012 (https://archive.ics.uci.edu/ml/datasets/Phishing+Websites#)

[4] R. M. Mohammad, F. Thabtah, and L. McCluskey. An assessment of features related to phishing websites using an automated technique. In 2012 International Conference for Internet Technology and Secured Transactions, pages 492–497, 2012.