Detection of Phishing Using Machine Learning

Prashantha G R [1], Meghana G R [2], Bhoomika P M [3], Prajwal C T[4], Vasu G [5], Shivasali Rajendra Babu[6]

Dept. of CSE, JIT, Davanegere, India

Abstract— **Phishing is known as social engineering crime/attack, a hacker or attacker tries to obtain important information from users and influence them by creating a harmful false link that perfectly resembles a real one. A person who clicks on the link is sent to the hacker's website rather than the legal one. User's data is compromised when they provide them sensitive information under the impression that it is authentic. As a result, there is a significant loss of personal information, and when an employee of the firm is compromised, there may also be a loss of corporate information. Due to this, there is a significant loss of one's personal data, and when an employee of the organization is compromised, there may also be a loss of the company's data. The traditional method of phishing detection involved creating a database containing a list of websites that should not be visited as well as the phishing links that were connected with it. This database was then compared to the entered URL to determine if it was or was not in the phishing database. However, there is a drawback to heuristic comprehensive search: if the phishing link is not in the database, there will be an issue and the user may still access the website. By using the website's link and validating them based on the URL characteristics, we may apply a machine learning method to get around the issue of exhaustive search and determine if a link is real or phished. We experimented with a number of methods before deciding to use the Random Forest algorithm on our dataset and developing a Chrome plugin.The random forest gives a accuracy of 89.60% and 0.59 seconds.**

Keywords: Phishing Recognition, Random Forest Algorithm, URL Feature Extraction, Web based Security, Chrome Plugin

I. INTRODUCTION

Phishing is the most dangerous criminal behaviour in the online world. Because the majority of consumers access government and financial institution services online, phishing efforts have expanded dramatically in recent years. Phishers began to generate money and are now running a profitable business. Phishers use a range of ways to target susceptible people, including SMS, VOIP, forged links, and bogus websites. It is fairly simple to create fake websites that mimic actual websites in terms of style and content. These websites would even have the same content as their original webpages. These websites are designed to acquire personal information from users, such as account numbers, login credentials, debit and credit card passwords, and so on.

Furthermore, attackers pose as high-level security measures, posing security questions for users to answer. Users who respond to those questions are more likely to become victims of phishing schemes. To combat phishing attacks, several groups throughout the world have been performing substantial study. Phishing attacks may be prevented by recognising phishing websites and teaching customers on how to recognise them. The use of machine learning algorithms has shown to be one of the most successful approaches for detecting phishing websites. This research explored many approaches for detecting phishing websites.

II. LITERATURE REVIEW

Alsaleh, M., Alarifi, A., and Al-Ghamdi, A. (2018). A unique anti-phishing extension based on machine learning algorithms. Future Internet, 10(8), 77.
This project provided a ground-breaking anti-phishing extension that utilised machine learning methods such as Random Forest and Support Vector Machine (SVM). The authors evaluated the extension's performance on a real dataset and discovered that it was quite good at identifying phishing.

R. Alhakami, I. Traore, and M. Oussalah. Detecting phishing assaults with machine learning techniques. IEEE Access, 7, pp. 74196-74208.
The authors provided a detailed assessment of phishing detection techniques, including ML-based algorithms. They reviewed several machine learning methods, Decision Trees, and Neural Networks used to detect phishing assaults. The poll emphasised the advantages and disadvantages of each algorithm in the context of phishing detection.

M. Alsulami, A. Alhaidari, M. Aldhahri, and L. Gao (2020). Machine learning algorithms are used to detect phishing emails. 3(2), 193-211, Journal of Cybersecurity and Privacy.
Using distinct variables derived from email headers and content, the authors compared the performance of multiple ML algorithms, including Logistic Regression, K-Nearest Neighbours, and Random Forest. The testing findings confirmed the efficacy of the proposed approach in detecting phishing emails.

Kumar, R., and P. Kumar (2021). A complete review of phishing detection using machine learning. 4(1), 20-33, Journal of Cybersecurity and Information Management.
The authors conducted a review of several machine learning approaches used in phishing detection. They talked about how to use various capabilities, such as URL-based. The review identified potential research directions as well as the strengths and limits of various methodologies.

P. Kumbhar and N. Jagtap (2022). a comprehensive review of the literature on machine learning-based phishing detection. 10–20 in the 14(3) issue of the International Journal of Intelligent Systems and Applications.
The purpose of this systematic literature review was to examine recent advances in phishing detection using machine learning approaches. The authors examined a variety of machine learning techniques, feature extraction methods, and datasets utilized in phishing detection studies. They also talked about the challenges and next steps for improving the effectiveness of ML-based phishing detection systems.

III. METHODOLOGY

A. Dataset

For our proposed system, we use a dataset obtained from the UCI machine learning repository. The phishing website attribute dataset consists of about eleven thousand URLs (examples), composed of about six thousand phishing cases and about five thousand real incidents. All of these examples have about twenty-five features, and each feature is associated with a set of decisions and follows a set of rules.

Table 1 demonstrates how the qualities and features are classified into several groups:

1) Address bar-based features
2) Anomalies-based features
3) JavaScript and HTML-based features
4) Domain system-based features.

Table 1: characteristics in dataset

@relation phishing
@attribute having_IP_Address (-1,1)
@attribute URL Length (1,0,-1)
@attribute having At Symbol (1,-1)
@attribute Prefix Suffix (-1,1}
@attribute having Sub Domain -1,0,1)
@attribute SSL final State -1,1,0)
@attribute Favicon [1,-1)
@attribute port (1,-1)
@attribute HTTPS token (-1,1)
@attribute Request URL (1,-1)
@attribute URL of Anchor (1,0,1)
@attribute Links in tags (1,-1,8}
@attribute SFH (-1,1,0)
@attribute Submitting to email (-1,1}
@attribute Abnormal URL -1,1
@attribute Redirect [8,1)
@attribute on mouseover {1,-1)
@attribute popUpWidnow 1,-1)
@attribute Iframe {1,-1 }
@attribute age of domain (-1,1)
@attribute web traffic (-1,0,1)
@attribute Page Rank -1,1 }

B. Features extraction

There are some traits and patterns that might be regarded as features on phishing websites. We attempt to cover all the aspects of phishing websites that have been used in prior studies in this area. When we looked at the phishing traits and patterns, we also found several new features that may be called characteristics. 32 phishing features are included in total, of which 3 are novel ones.

- The URL can be used to extract features.

- Content from the page can be used to extract features.

- The page rank can be used to extract features.

Because the purpose of phishing websites is to steal sensitive information like passwords and credit card numbers, we use the number of emails entered and the number of passwords entered as additional features.

C. Pre-processing

A dataset taken from the UCI repository is used by the phishing detection plugin. The dataset contains 30 characteristics that must be reduced before they can be extracted on the browser without the need of external online services or third-party software. To do this, each feature is evaluated on the browser to assess its extraction feasibility. While additional features can improve accuracy, they can also slow down feature extraction rates, making it more difficult to identify phishing quickly.Thus, a subset of features is chosen to balance the tradeoff between accuracy and speed.

Eliminating Duplicates: Duplicate data in a dataset can skew machine learning model training and cause overfitting. The plugin thereby eliminates any redundant entries from the dataset.

handling missing values or NaN values: that may be included in the dataset. The median of the related feature values in the dataset is used by the plugin to fill in any missing values.

categorical features: In order for machine learning algorithms to process they may need to be encoded as numerical values in the dataset. The plugin uses one-hot encoding to encode categorical information, with each category being represented by a binary value.

Data scaling: The dataset may include characteristics with a variety of value ranges. The data can be scaled to a common range to speed convergence and improve forecast accuracy. The plugin uses the Scikit-learn library's Standard Scaler to scale the dataset.

Feature Selection: As previously indicated, the plugin reduces the original dataset's 30 features to just 17 for effective browser extraction. The machine learning model is then trained using the chosen features.

Table 2: Webpage Features

IP addresses	Degree of subdomain	Anchor tag href domains
URL length	HTTPS	Script & link tag domains
URL shortener	Favicon domain	Empty server from handler
@' in URL	TCP Port	Use of mailto
Redirection with '//'	HTTPS in domain name	Use of iFrame,
-' in domain	Cross domain requests	

The dataset is then divided into a training set and a testing set, with a 30% testing set. The testing and training data are also stored on disc.

D. Training

The training data for the preprocessing module is imported from disc. On the data, a random forest classifier is trained using the scikit-learn toolkit. Random Forest utilises a set of ten decision tree estimators as an ensemble learning technique(1). Every decision tree applies the CART algorithm and strives to reduce gini impurity.

$\text{Gini}(E) = 1 - c \sum j=1\ p2$--------(1)

Additionally, the cross-validation score is created using the training data. The F1 score is computed based on the test results. Finally, the trained model is exported to JSON in the succeeding module.

E. Plugin Feature Extraction

For every webpage, 32 highlights must be extracted and encoded in real-time as the page is being stacked.

The script uses a substance to access the DOM of the webpage. Each page thus receives the substance script when it loads. The highlight data can be gathered by the content script and sent to the plugin afterwards. The main goal of this job is to avoid using any other online services, and the highlights must be free of organise inactivity and extraction must be quick. All of these are proven without a shadow of a doubt when technologies for highlight extraction are developed. When a highlight is extracted, it is encoded into the following values: -1, 0, 1.

-1 - Genuine

0 - Suspicious

1 – Phishing

F. System Design

Using Python scikit-learn, an Arbitrary Random Forest classifier is trained on a dataset of phishing sites. A JSON structure has been created to communicate with the arbitrary timberland classifier, and the learnt classifier has been exchanged to it. A browser script that employs the trading model has been executed.

JSON is used to categorise the site that is being layered within the dynamic browser tab. The framework warns the client of the possibility of phishing. The arbitrary Woodland classifier on 32 website features is used to determine if the destination is phishing or legitimate. The dataset record is stacked using the Python library, and 17 highlights are picked from a selection of 32. Highlights are chosen with the assumption that they can be extracted totally offline on the client side without relying on a web service or a third party. The dataset with the selected highlights is then segregated for preparation and testing. The Arbitrary Woodland is then prepared on the training data and supplied to the above defined JSON arrangement. A URL is provided to facilitate the JSON record. The client-side Chrome plugin is designed to run a script on each stack of pages, at which point it starts to extract and encrypt the selected highlights. After the highlights have been decoded, the plugin then checks to see if the trading show JSON is cached and downloads it again if it isn't.

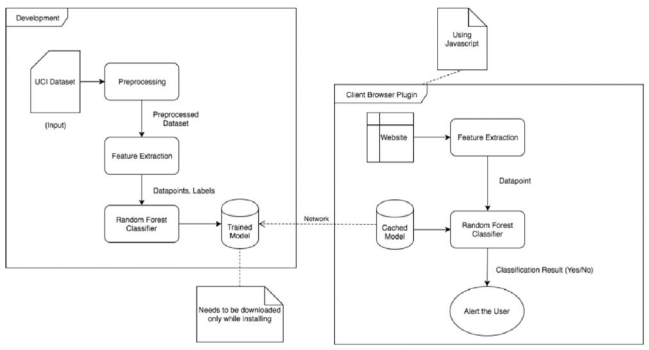


Fig 1: System Architecture

G. Random Forest

Popular machine learning algorithm Random Forest may be used to identify phishing websites. It is an ensemble learning technique that builds numerous decision trees and produces a class that is the mean of the classifications or mean prediction of the individual trees (regression).

Random Forest has several advantages that make it suitable for phishing detection, including:

1. Ability to handle high-dimensional data: Random Forest can handle datasets with a large number of features, such as those commonly used in phishing detection.

2. Robustness to noise and outliers: Random Forest is less sensitive to noisy data and outliers than other machine learning algorithms, making it effective in detecting malicious websites that may have unexpected characteristics.

3. Good precision: Random Forest often yields precise answers, especially when the forest's density of trees is large.

4.Reduced danger of overfitting: Compared to other decision tree-based algorithms, Random Forest is less prone to overfitting..

In the detection of phishing websites, Random Forest can be trained on a dataset of known phishing and legitimate websites, using various features such as URL length, presence of certain keywords, and SSL certificate information. The trained model can then be used to predict the class of new websites and determine whether they are legitimate or phishing.

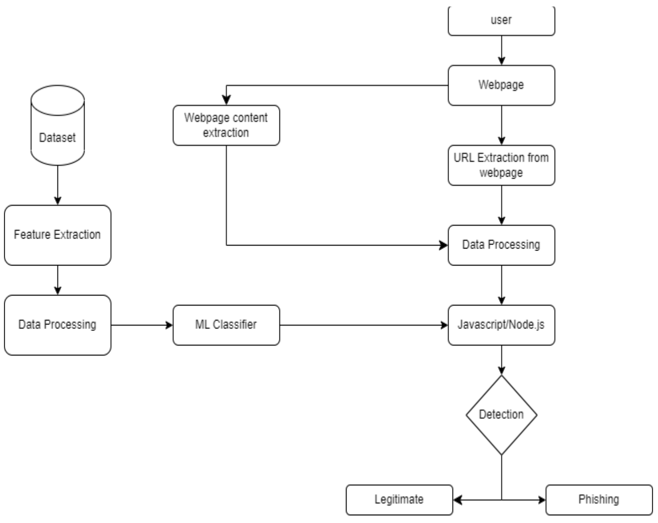


Fig 2: Flow Chart

H. Class Diagram

This Figure displays the class diagram for the full machine translation system. This picture clearly illustrates the various system modules' functions. Additionally, it demonstrates how the system's components interact with one another, giving implementers a clear notion of how to proceed.

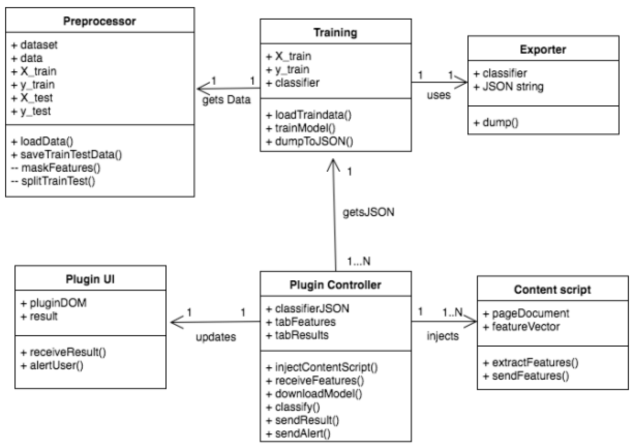


Fig 3: Class Diagram

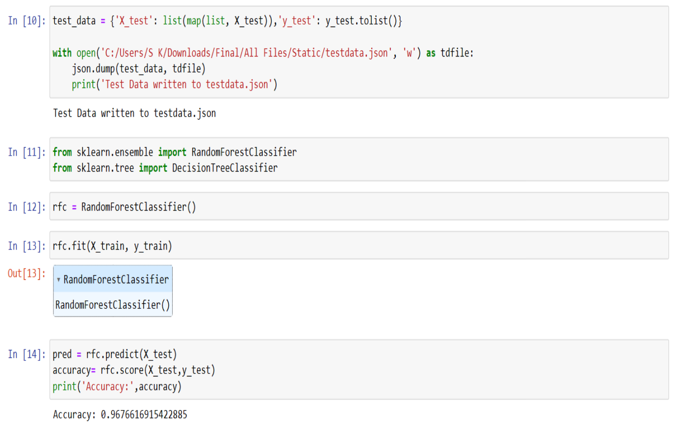
IV RESULTS

A. DATASET FOR TESTING

The test set comprises of data points that are 70:30 divided from the dataset. Additionally, phish Tank-listed URLs are used to test the plugin. Additionally, as soon as they are discovered, new phishing websites are added to PhishTank. The plugin can also identify new phishing websites, it should be emphasized. Below is a summary of the findings from the testing of both this module and the complete system.

B. PRE-PROCESSING

Preprocessing model output for detection of Phishing Websites

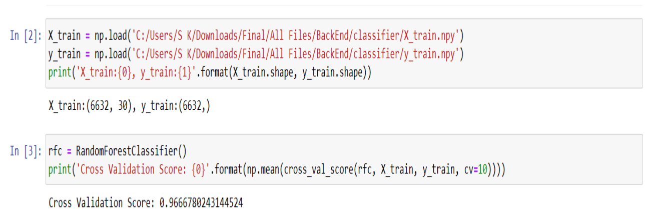


```
In [10]: test_data = {'X_test': list(map(list, X_test)),'y_test': y_test.tolist()}

         with open('C:/Users/S K/Downloads/Final/All Files/Static/testdata.json', 'w') as tdfile:
             json.dump(test_data, tdfile)
             print('Test Data written to testdata.json')

         Test Data written to testdata.json

In [11]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.tree import DecisionTreeClassifier

In [12]: rfc = RandomForestClassifier()

In [13]: rfc.fit(X_train, y_train)

Out[13]: ▾ RandomForestClassifier
         RandomForestClassifier()

In [14]: pred = rfc.predict(X_test)
         accuracy= rfc.score(X_test,y_test)
         print('Accuracy:',accuracy)

         Accuracy: 0.9676616915422885
```

Fig 4: Pre-processing

C. TRAINING

Training Model Output after the pre-processing model



```
In [2]: X_train = np.load('C:/Users/S K/Downloads/Final/All Files/BackEnd/classifier/X_train.npy')
        y_train = np.load('C:/Users/S K/Downloads/Final/All Files/BackEnd/classifier/y_train.npy')
        print('X_train:{0}, y_train:{1}'.format(X_train.shape, y_train.shape))

        X_train:(6632, 30), y_train:(6632,)

In [3]: rfc = RandomForestClassifier()
        print('Cross Validation Score: {0}'.format(np.mean(cross_val_score(rfc, X_train, y_train, cv=10))))

        Cross Validation Score: 0.9666780243144524
```

Fig 5: Training Output

D. CLASSIFICATION

Using HTML and CSS, a user-friendly User Interface has been created for the plugin. A sizable circle in the user interface (UI) shows what proportion of the webpage in the active tab is legitimate. In accordance with the classification output, the circle's color also changes. The analysis's findings, which include the features that were extracted, are shown below the circle in the following color scheme.

- Yellow - Suspicious
- Green - Valid
- Red Light – Phishing

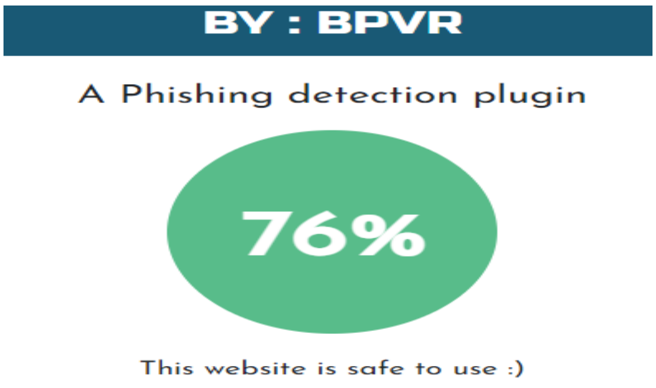


Fig 6: safe to use

E. FEATURES THAT USED FOR DETECTION

The Features That Are Used for Detection of Phishing Websites. For example, URL Length, @ Symbol, Prefix/Suffix In The Domain, Redirecting Using//, Anchor And SFH Etc.

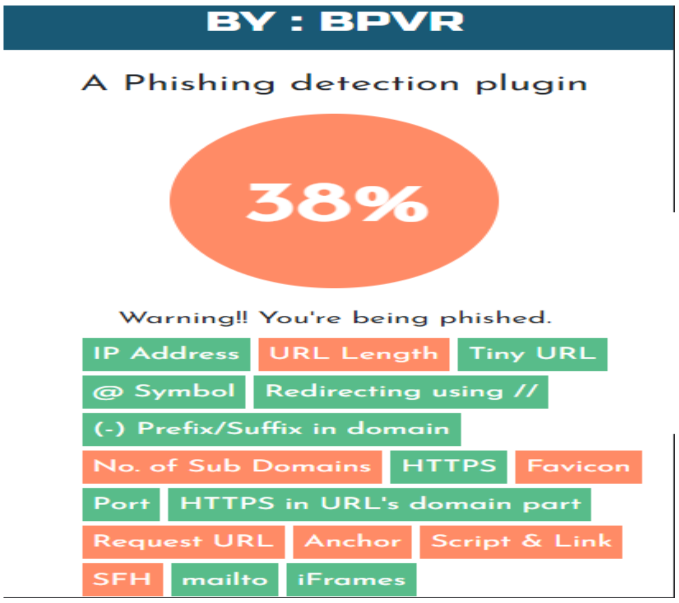


Fig 7: Features used for Detection

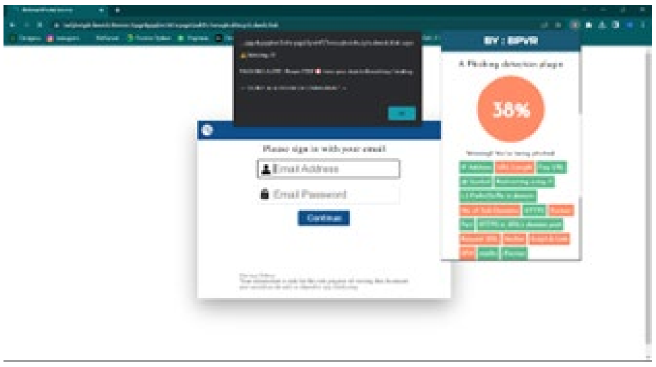


Fig 8: Sample screenshot of the blocking of phishing website

V. EXPECTED OUTCOMES

The outcome of using the phishing-detection-plugin to detect phishing websites is to provide an additional layer of protection for users while they are browsing the web. The plugin can identify and alert users to potential phishing websites, allowing them to avoid visiting or interacting with these sites.

By using the plugin, users can reduce their risk of falling victim to phishing scams, which can result in stolen personal information, financial losses, or other types of fraud. The plugin can also help users to recognize the signs of phishing attacks and improve their overall awareness of online security risks.

Overall, the outcome of using the phishing-detection-plugin should be a safer and more secure browsing experience for users, with increased protection against phishing attacks and other types of online scams.

VI. CONCLUSION

Phishing, a social engineering assault, has a significant negative influence on society at large by destroying the financial and economic worth of businesses, government agencies, and people by using various phishing strategies to exploit their personal information.

By utilizing a machine learning technique, we have presented a Chrome plugin that automates the process of identifying and defeating conventional tactics. After reviewing a number of alternative algorithms, we chose random forest as our primary method utilizing the k-fold validation technique based on the confusion matrix and execution rate. The accuracy and runtime of the random forest approach are 89.60% and 0.59 seconds, respectively.

REFRENCES

[1] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, “Intelli- gent phishing website

detection using random forest classifier,” 2017 International Conference on Electrical and

Computing Tech- nologies and Applications (ICECTA), Nov. 2017.

[2] “UCI Machine Learning Repository: Phishing Websites Data Set,” [Online]. Available:

https://archive.ics.uci.edu/ml/datasets/ phishingwebsites.

[3] J.-H. Li and S.-D. Wang, “PhishBox: An Approach for Phishing Validation and

Detection,” 2017 IEEE 15th Intl Conf on Depend- able, Autonomic and Secure Computing,

15th Intl Conf on Perva- sive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence

and Computing and Cyber Science and Technology Con-gress

(DASC/PiCom/DataCom/CyberSciTech), 2017.

[4] A. A. Ahmed and N. A. Abdullah, “Real time detection of phishing websites,” 2016 IEEE

7th Annual Information Technology, Elec- tronics and Mobile Communication Conference

(IEMCON), 2016.

[5] R. Aravindhan, R. Shanmugalakshmi, K. Ramya, and S. C., “Cer- taininvestigation on web

application security: Phishing detection and phishing target discovery,” 2016 3rd International

Conference onAdvanced Computing and Communication Systems (ICACCS), 2016.

[6] S. B. K.p, “Phishing Detection in Websites Using Neural Networks and Firefly,”

International Journal Of Engineering And Computer Science, 2016.

[7] “An Efficient Approaches For Website Phishing Detection Using Supervised Machine

Learning Technique,” International Journal of Advance Engineering and Research

Development, vol. 2, no. 05, 2015.

[8] S. Gupta and A. Singhal, “Phishing URL detection by using artifi-cial neural network

with PSO,” 2017 2nd International Conference on Telecommunication and Networks (TELNET),

2017.

[9] Ammar ALmomani, G. B. B, Tat-Chee Wan, Altyeb Altaher, and Selvakumar Manickam,

“Phishing Dynamic Evolving Neural Fuzzy Framework for ...,” Jan-2013. [Online]. Available:

https://arxiv.org/ pdf/1302.0629.