A Google Extension for Phishing Websites Detection Using Random Forest

Xuemei Shang
Department of Computer Science
New York Institute of Technology
Vancouver, Canada
xshang@nyit.edu

Ming Jing
Department of Computer Science
New York Institute of Technology
Vancouver, Canada
mjing01@nyit.edu

***Abstract*—The phishing attack is one of the most prevalent cybercrime techniques, which is carried out by imitating a legitimate website to get personal information from internet users. Machine learning takes advantage of computer algorithms to automate analytical model building through experience. This paper analyzed four papers related to machine learning algorithms for phishing detection. Based on this research, this project explored three machine learning models, including Random Forest, SVM, and J48, for analyzing a public UCI phishing dataset by using WEKA. Moreover, this project presented a detailed analysis of phishing attacks and evaluated the above machine learning approaches. Finally, this project developed a Chrome extension with the most outperformed algorithm - Random Forest to detect phishing websites.**

Keywords—Phishing, Machine Learning, SVM, J48, Random Forest, Google Extension

I. Problem Statement

Currently, with the popularity of online transaction and remote working during COVID-19, phishing has become one of the most prevalent threats to millions of internet users. According to the Phishing Landscape 2021 report[1], from 1 May 2020 through 30 April 2021, they collected 1,487,914 phishing reports from four widely used threat intelligence providers: the Anti-Phishing Working Group(APWG), OpenPhish, PhishTank, and Spamhaus. The report shows phishing increased by nearly 70% during this period and continues to become a significant threat to companies, online service providers, and internet users.

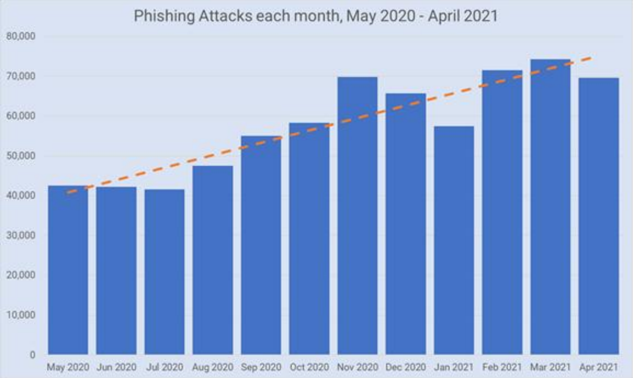


Fig. 1. Monthly Number of Phishing Attacks, 1 May 2020 to 30 April 2021[1]

In a phishing attack, the attacker obtains client-sensitive data (such as account numbers, login passwords, insurance numbers, credit card details, etc.) using forged emails and masqueraded websites. It is more crucial than ever to prevent users from being attacked by accurately detecting phishing. Recent study demonstrates that machine learning algorithms can detect phishing websites effectively, with better accuracy than existing approaches like Blacklist Approach and Whitelist Approach[2]. This project compared the performance of different machine learning algorithms, and then using the best classifiers in our Chrome extension to provide a real-time phishing detection.

II. Related Work

A. Background

Because phishing attacks are an increasing trend, previous papers have proposed various approaches to solve this problem. According to[3], there are five anti-phishing categories:

- Heuristic-based approach
- Content-based approach
- Blacklist based approach
- Machine learning approach,
- Hybrid combination of different techniques approach

However, the approaches with the combination of machine learning and hybrid techniques of different machine learning algorithms are more effective and intelligent[2]. Nowadays, machine learning algorithms are widely used for phishing detection in many papers.

B. Related papers

In [4], the authors applied the C4.5 decision tree model to their dataset, containing 1000 web pages and eight predefined attributes. The dataset was collected through a Chrome browser plugin developed by themself. They achieved an overall accuracy of 93.2%, and they developed a chrome extension with the C4.5 model to detect phishing websites. This paper proposed using machine learning for phishing detection and implementing it with a chrome plugin.

In [5], the authors used the C4.5 and CART algorithms to determine the effect of feature selection on the phishing dataset. The dataset was collected from the UCI machine learning repository database[16]. They showed that the CART algorithm with an accuracy of 94.4% is better than the C4.5

algorithm with an accuracy of 94.3%. This paper demonstrated that CART is a little better than C4.5 in phishing detection but without implementation in an actual application.

In [6], the authors tested different classifiers, including ANN, SVM, C4.5, RF, etc., to construct an intelligent system for phishing website detection. This paper also used the publicly available phishing dataset from the UCI machine learning repository[17]. Results showed that Random Forest outperformed the classification methods by achieving the highest accuracy of 97.36%. This paper showed that the Random Forest algorithm is much better than other algorithms, including ANN, SVM, and C4.5.

In [7], the authors applied machine learning techniques and algorithms for detecting phishing websites and achieved the following accuracy results: Logistic Regression(96.23%), Decision Tree(96.23%), Random Forest(96.58%). This paper analyzed phishing features based on phishing and legitimate websites scraped over from the internet. This paper showed that Random Forest indeed performed well in phishing detection. But both the two papers [6] and [7] didn't implement phishing detection in an actual application.

Based on the research of these papers, many machine learning algorithms have been used to detect phishing attacks, including C4.5, CART, ANN, SVM, RF, LR, etc. The purpose of using machine learning for phishing detection is promising and effective. This project focused on machine learning approaches for phishing attack detection and implement a chrome extension with the most outperformed algorithm to detect phishing websites.

III. Proposed Solution

Based on the previous research on machine learning techniques focusing on phishing attack detection, this project aims to apply some machine learning algorithms, including Support Vector Machines(SVM), Random Forest, and J48, to detect phishing websites and to achieve higher accuracy. The dataset for this project is from the UCI machine learning repository[12]. This project performed phishing features analysis and extraction of this dataset because phishing features are the key characteristics to determine whether a website is malicious or not. Then split the dataset into a training dataset and a testing dataset. Training and testing the dataset and comparing and evaluating the performance of the above classification algorithms, then concluded the best algorithm appropriate for phishing detection. Finally, this project implemented a chrome extension with the best-optimized classifier for detecting phishing websites.

IV. Methodology

This section discusses the proposed phishing detection methodologies. This project focuses on the following basic concepts: machine learning, phishing, SVM, Random Forest, and J48.

A. Machine Learning

Machine learning is a sub-field of artificial intelligence. It specializes in analyzing and interpreting the structure of data to achieve the purpose of learning, reasoning, and decision-making without manual interaction. Machine learning supports users to input a large amount of data into computer algorithms. Then let the computer analyze these data and give data-driven suggestions and decisions only based on the input data[8].

B. Phishing

Phishing is an attack that attempts to steal sensitive personal information, such as passwords, credit card details, bank information, etc., by masquerading websites that pretend to be legitimate[9].

C. Support Vector Machines

Support vector machines (SVMs) algorithm is a classification method for linear and nonlinear data. It uses nonlinear mapping to transform the original training data into higher dimensions. The new dimension is used to search the linear optimal separation hyperplane. By properly nonlinear mapping to a high enough dimension, the two types of data can always be separated by a hyperplane. SVM uses support vectors and margins to find this hyperplane[10].

D. Random Forest

The random forest adopts the integrated learning method - the divide and conquer method to improve its performance. In a simple decision tree, input is added to the root and then traversed down to the tree, resulting in a smaller subset. The integration mechanism combines various subsets of random trees in the random forest. Enter or test to traverse all trees. The final result is calculated according to the average value of individual results or the voting majority of classified data. Random forests are used for classification, regression, and other tasks, which are performed by building a large number of decision trees during training [10].

E. J48

The C4.5 algorithm is a classification algorithm to produce decision trees. J48 is an implementation of the C4.5 algorithm and has many additional features: calculation of missing values, decision tree pruning, continuous attribute value ranges, rules derivation, etc. J48 is an open-source Java implementation of the C4.5 algorithm in Weka[11].

V. Dataset Analysis

A. Dataset Introduction

The dataset of this project is collected from the UCI machine learning public repository[12]. This dataset was also used for machine learning-based phishing detection in other papers[6][13]. The details of the dataset are as follows:

Dataset	Number of Features	Number of Instances	Phishing Websites	Non-Phishing Websites
UCI[12]	30	11055	6157	4898

B. Dataset features analysis

Whether a website is malicious or not depends on its features. So it is important to analyze these features. Basically, these features could be divide into four categories[18]:

- Address bar based features (12 features)

F0: Having IP Address

F1: Hiding the Suspicious Part in Long URL

F2: Shortening Services in URL("TinyURL")

F3: Having "@" Symbol in URL

F4: Using "//" to Redirect

F5: Separating domain with adding Prefix or Suffix by (-)

F6: Having Sub Domain and Multi Sub Domains

F7: HTTPS

F8: Domain Registration Length

F9: Favicon

F10: Using Non-Standard Port

F11: "HTTPS" in the Domain Part of the URL

- Abnormal based features (6 features)

F12: Request URL

F13: URL of Anchor

F14: Links in tags

F15: Server Form Handler

F16: Submitting Information to Email

F17: Abnormal URL

- HTML and JavaScript based features (5 features)

F18: Website Forwarding

F19: Status Bar Customization

F20: Disabling Right Click

F21: Using Pop-up Window

F22: IFrame Redirection

- Domain based features (7 features)

F23: Age of Domain

F24: DNS Record

F25: Website Traffic

F26: PageRank

F27: Google Index

F28: Number of Links Pointing to Page

F29: Statistical-Reports Based Feature

VI. Classification Experiment

1) Tool

This project conducted experiments using the well-known Waikato Environment for Knowledge Analysis (WEKA), a data mining tool with a collection of machine learning algorithms to deal with data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.

2) Hardware

- 64-bit Windows 10
- Hard disk of at least 64 GB
- RAM at least 8GB

A. Test Procedure Design

Fig2 shows an architecture of proposed work procedure of this project.

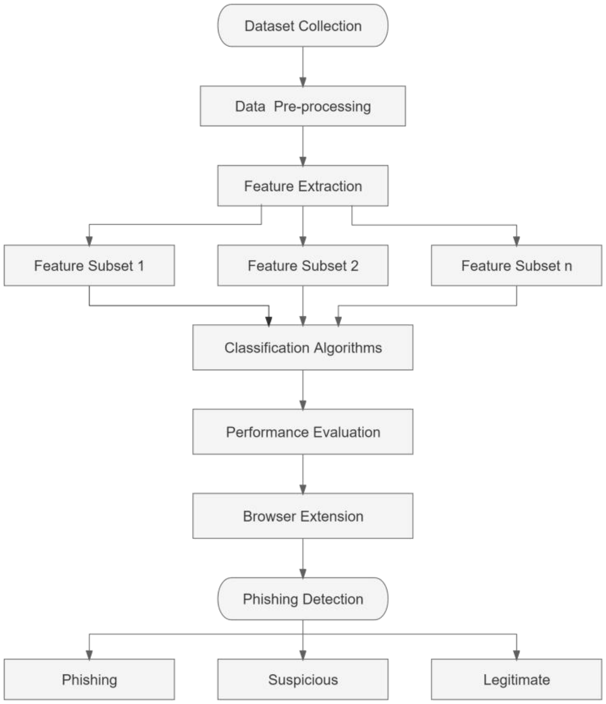


Fig. 2. Flowchart of proposed work using Machine Learning

B. Result Analysis of Test Procedure Performed

This project performed experiments with three classifiers: Random Forest, SVM, and J48. two test options were applied with these classifiers: Cross-validation(10 folds) and dataset percentage split(70:30, 80:20, 90:10). The experiment results showed in Table 1 and Table 2. Fig 3 shows the bar chart comparison result of these classifiers.

TABLE I. Classifier's performance: Cross-validation

Classifier	Accuracy	Precision	Recall	F-Measure
RF	97.26%	97.3%	97.3%	97.3%
SVM	94.50%	94.5%	94.5%	94.5%
J48	95.87%	95.9%	95.9%	95.9%

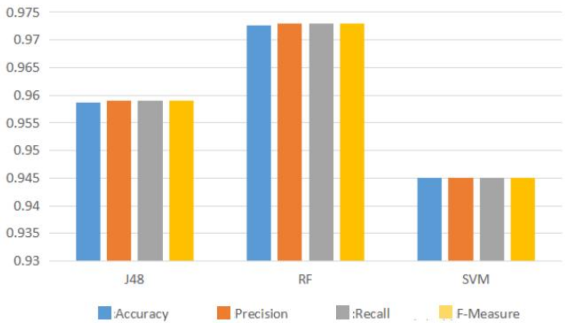


Fig. 3. Result Comparison

TABLE II. CLASSIFIER'S PERFORMANCE: DATASET PERCENTAGE SPLIT

Dataset Split	Classifier	Accuracy	Precision	Recall	F-Measure
70:30	RF	97.22%	97.2%	97.2%	97.2%
	SVM	94.33%	94.4%	94.3%	94.3%
	J48	95.59%	95.6%	95.6%	95.6%
80:20	RF	97.15%	97.2%	97.2%	97.1%
	SVM	94.53%	94.6%	94.5%	94.5%
	J48	96.11%	96.11%	96.11%	96.11%
90:10	RF	97.29%	97.3%	97.3%	97.3%
	SVM	94.84%	95.0%	94.8%	94.8%
	J48	95.93%	96.0%	95.9%	95.9%

C. Discussion of Project and Results

This project aims to identify phishing URLs by looking into the classifiers' accuracy, precision, recall, and F-Measure of machine learning algorithms, including Random Forest(RF), Support Vector Machines(SVM), and Decision Tree(J48). Through the experiments, cross-validation with 10 folds takes more time to build a model than the dataset percentage split approach. From Table 2, splitting the dataset into two parts: 90% for training and 10% for testing, could achieve a better result than the other two percentage split methods. Random Forest could obtain higher accuracy in above all experiments and achieve an accuracy of 97.29% by using the 90:10 percentage split approach.

VII. Feature Selection

Feature selection is the method of reducing the dimensions of input variables by using only relevant data to avoid noise in data. It is the process of choosing relevant features to improve the performance and reduce the computational cost of the machine learning model. In Weka, there are two parts for feature selection: attitude evaluator and search method. The attribute evaluator is a technique to evaluate each feature of a dataset in the context of the output variable. The search method is a technique to try different combinations of dataset features to obtain a shortlist of chosen attributes[19].

This project applied the combination of filter attribute evaluator and ranker search method for feature selection. The filter attribute evaluator measures the relevance of features by their correlation with the dependent variable, and the ranker search method ranks attributes by their individual evaluations.

A. CorrelationAttributeEval + Ranker

CorrelationAttributeEval evaluates the worth of an attribute by measuring the correlation between it and the class.

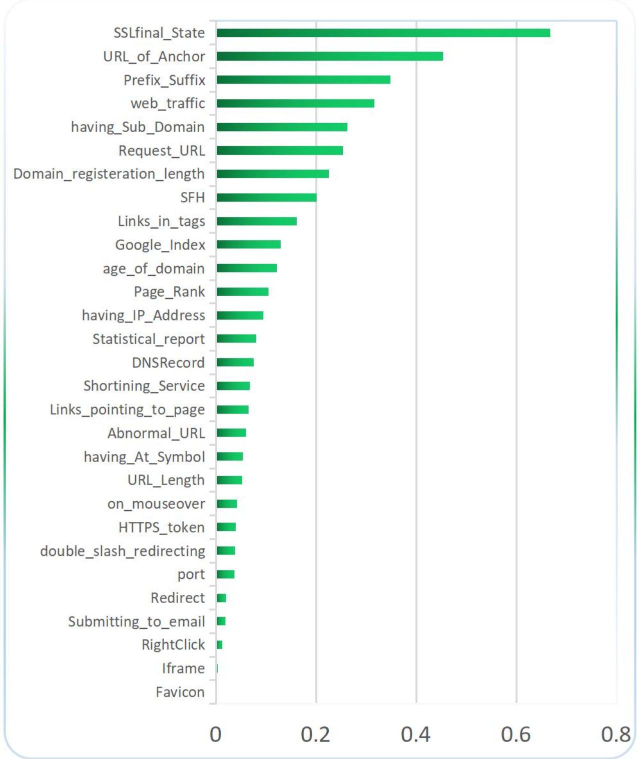


Fig. 4. CorrelationAttributeEval + Ranker

B. GainRatioAttributeEval + Ranker

GainRatioAttributeEval evaluates the worth of an attribute by measuring the gain ratio with respect to the class.

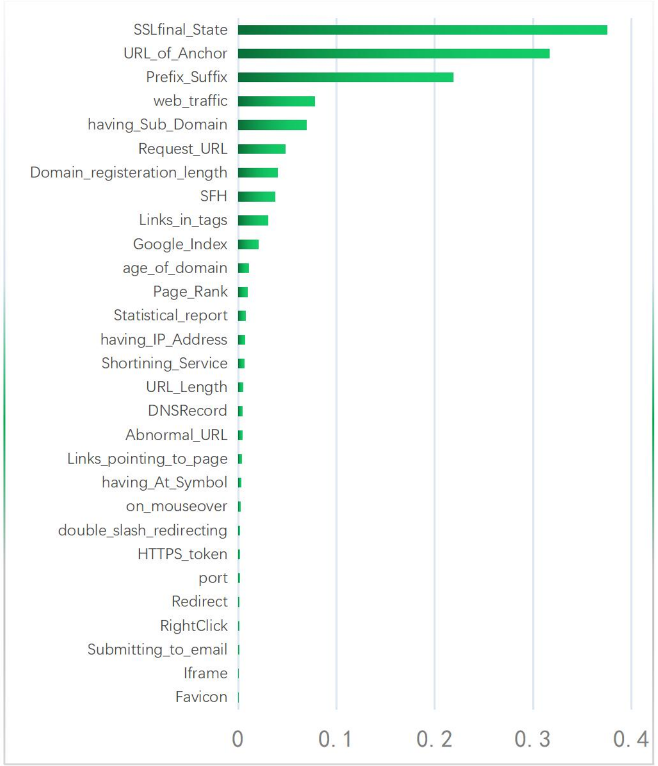


Fig. 5. GainRatioAttributeEval + Ranker

C. InfoGainAttributeEval + Ranker

InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class.

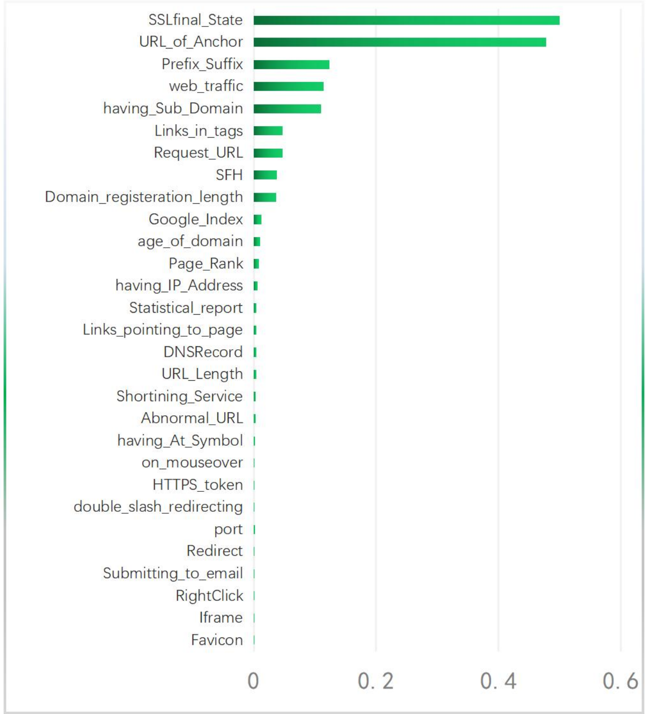


Fig. 6. InfoGainAttributeEval + Ranker

D. SymmetricalUncertAttributeEval + Ranker

SymmetricalUncertAttributeEval evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

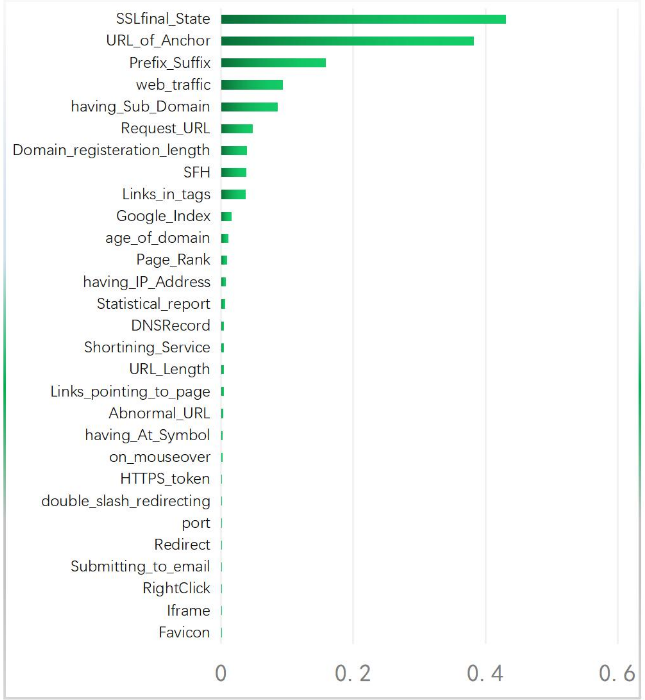


Fig. 7. SymmetricalUncertAttributeEval + Ranker

E. Result Analysis of Feature Selection

1) Top 23 important features

F7	SSLfinal_State
F13	URL_of_Anchor
F5	Prefix_Suffix
F25	web_traffic
F6	having_Sub_Domain
F12	Request_URL
F8	Domain_registeration_length
F15	SFH
F14	Links_in_tags
F27	Google_Index
F23	age_of_domain
F26	Page_Rank
F0	having_IP_Address
F29	Statistical_report
F24	DNSRecord
F2	Shortining_Service
F28	Links_pointing_to_page
F17	Abnormal_URL
F3	having_At_Symbol
F1	URL_Length
F19	on_mouseover
F11	HTTPS_token
F4	double_slash_redirecting

2) Top 7 less important features

F10	port
F18	Redirect
F16	Submitting_to_email
F20	RightClick
F22	Iframe
F11	Favicon
F21	popUpWidnow

F. Discussion of Project and Results

Section VI-C concluded that Random Forest could achieve an accuracy of 97.29% by using the 90:10 percentage split approach without feature selection of the dataset. Based on the above result of feature selection, performed the Random Forest classification with the 90:10 dataset split percentage again with the top 23 important features, we could get the following results:

TABLE III. CLASSIFICATION RESULTS WITH FEATURE SELECTION

Classifier	Accuracy	Precision	Recall	F-Measure
RF(30 features)	97.29%	97.3%	97.3%	97.3%
RF(23 features)	97.46%	97.5%	97.5%	97.5%

The above experiment showed that by performing feature selection appropriately, the classification accuracy could improve, and the accuracy of the Random Forest classifier improved to 97.46% after feature selection. Feature selection is essential, especially in a dataset with many features. The more features, the more complex the model, and the longer time it takes to analyze features and train models.

VIII. Google Extension Implementation

Based on the previous experiment, the Random Forest algorithm could achieve a better classification accuracy than other algorithms in phishing detection with the given dataset. Therefore, this project developed a Google extension using the Random Forest algorithm.

A. *Plugin System Design*

The backend applied python to use the Random Forest algorithm to train the model with the UCI dataset. The frontend requested the trained model to classify the current website URL features and show phishing or safe of this website.

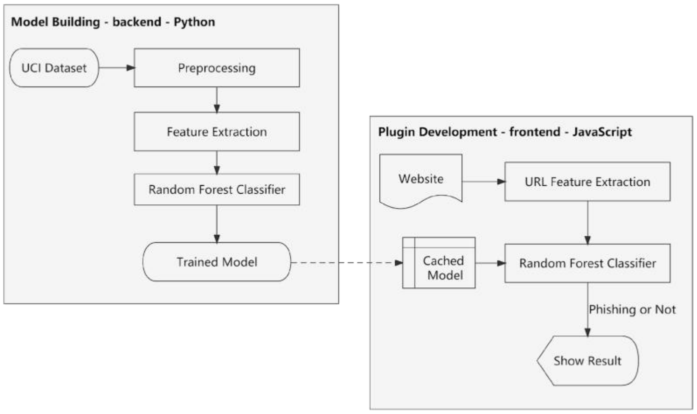


Fig. 8. Plugin System Design

B. *Plugin System Sequence*

The plugin system sequence showed in Fig 9:

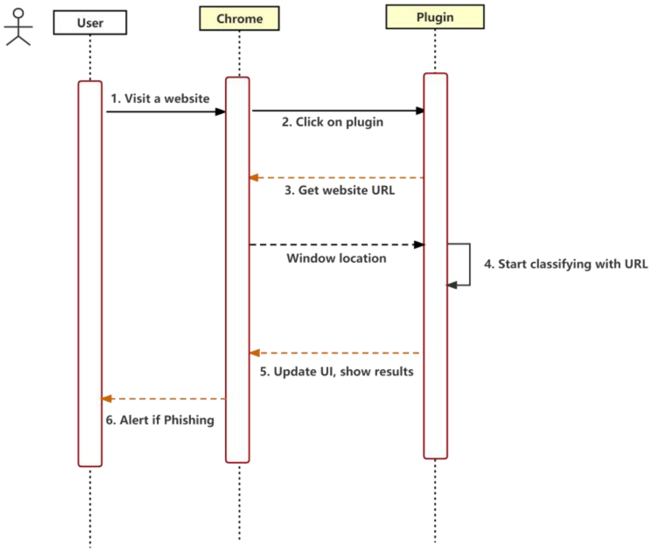


Fig. 9. Plugin System Sequence

C. *Plugin Code Structure*

The plugin code structure showed in Fig 10:

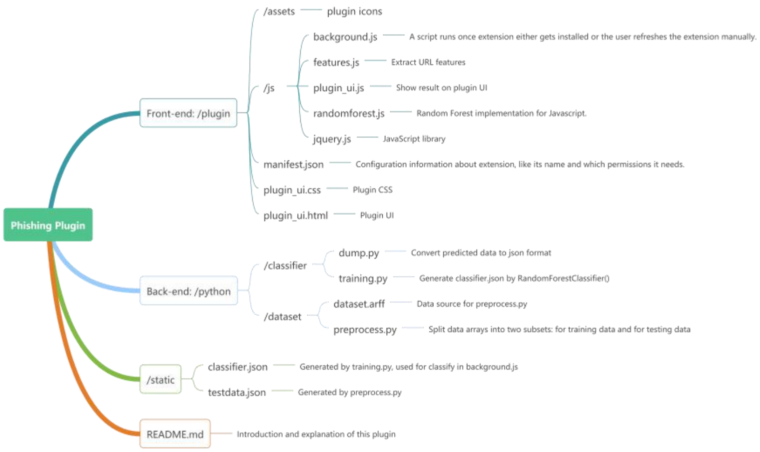
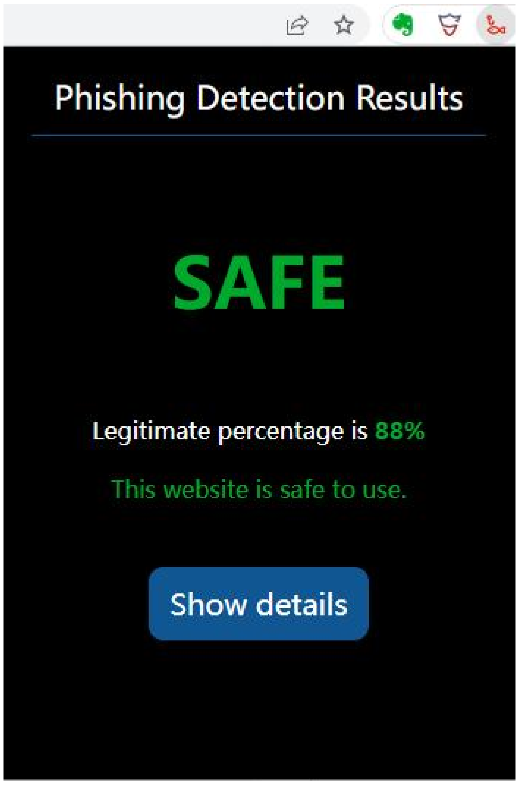

Phishing Plugin
Front-end: /plugin
/assets
plugin icons
/js
background.js
A script runs once extension either gets installed or the user refreshes the extension manually.
features.js
Extract URL features
plugin_ui.js
Show result on plugin UI
randomforest.js
Random Forest implementation for Javascript.
jquery.js
JavaScript library
manifest.json
Configuration information about extension, like its name and which permissions it needs.
plugin_ui.css
Plugin CSS
plugin_ui.html
Plugin UI
Back-end: /python
/classifier
dump.py
Convert predicted data to json format
training.py
Generate classifier.json by RandomForestClassifier()
/dataset
dataset.arff
Data source for preprocess.py
preprocess.py
Split data arrays into two subsets: for training data and for testing data
/static
classifier.json
Generated by training.py, used for classify in background.js
testdata.json
Generated by preprocess.py
README.md
Introduction and explanation of this plugin

Fig. 10. Plugin Code Structure

D. *Plugin Detection Result*

1) *Safe Website*

If a website is safe, when you click the plugin icon, the plugin would popup with safe(Fig 11).



Phishing Detection Results

SAFE

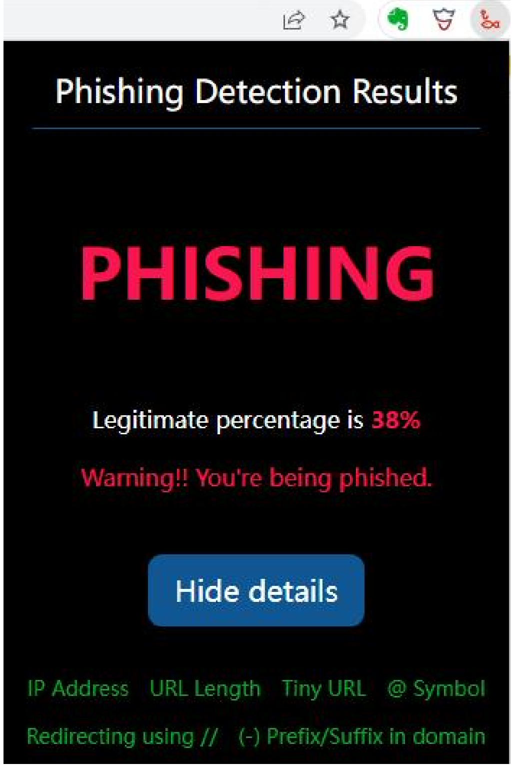
Legitimate percentage is 88%

This website is safe to use.

Fig. 11. Safe Website

2) *Phishing Website*

If a website is phishing, when you click the plugin icon, the plugin would pop up with phishing(Fig 12). And when you click the show details button, detailed features of the current website URL shows below, features in color green are legitimate, features in color white are suspicious, and features in color red are phishing(Fig 13).

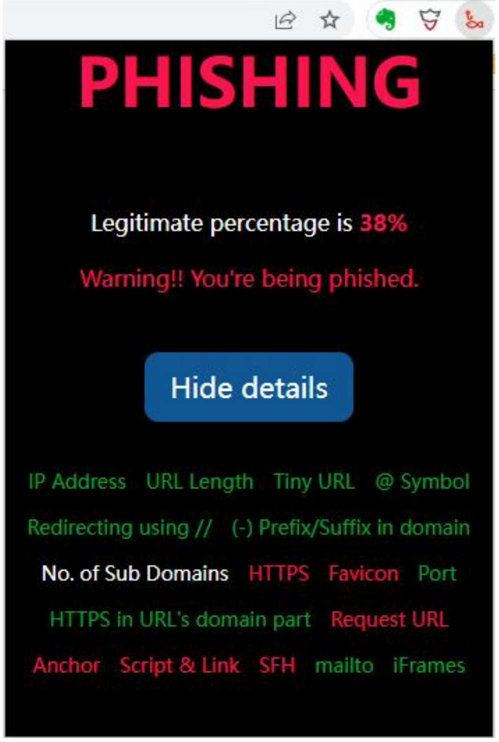

Phishing Detection Results

PHISHING

Legitimate percentage is 38%

Warning!! You're being phished.

Fig. 12. Phishing Website


PHISHING

Legitimate percentage is 38%

Warning!! You're being phished.

Fig. 13. Feature Details

IX. CONCLUSION

Phishing attacks are an increasingly significant threat posed to internet users. The attackers steal victims' personal information via spoofing emails or masquerading websites. Phishing websites detection is much more crucial than ever before because more and more people have started online shopping or remote working since COVID-19. Random Forest is an efficient machine learning approach due to its high classification accuracy and classification speed. This project summarized previous related research that analyzed the phishing dataset features and rules to define a malicious website. In this project, several machine learning algorithms were performed and concluded that the Random Forest classifier is better than other classifiers regarding classification accuracy, precision, and F-Measure.

Furthermore, this project tried feature selection, obtained the importance rank of all these features, and showed that appropriate feature selection improves classification accuracy. Finally, this project implemented a Google extension with the Random Forest algorithm. The test result showed that this extension could detect most phishing websites accurately but failed to detect all phishing websites. There are some limitations to extracting some features accurately, such as domain expiration time, registration time, abnormal URL, etc. The WHOIS API service is not free, and some free API is not stable and not accurate enough. However, this project is meaningful and helpful in identifying some phishing websites.

REFERENCES

[1] Interisle Consulting Group - Insights: Whitepapers. (2021b). Greg Aaron. https://www.interisle.net/PhishingLandscape2021.html

[2] V. Patil, P. Thakkar, C. Shah, T. Bhat, and S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5.

[3] S. Patil and S. Dhage, "A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 588-593, doi: 10.1109/ICACCS.2019.8728356.

[4] J. Bohacik, I. Skula and M. Zabovsky, "Data Mining-Based Phishing Detection," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), 2020, pp. 27-30, doi: 10.15439/2020F140.

[5] R. Wahyudi, H. Marcos, U. Hasanah, B. P. Hartato, T. Astuti and R. A. Prasetyo, "Algorithm Evaluation for Classification "Phishing Website" Using Several Classification Algorithms," 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE), 2018, pp. 265-270, doi: 10.1109/ICITISEE.2018.8720975.

[6] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2017, pp. 1-5, doi: 10.1109/ICECTA.2017.8252051.

[7] V. Patil, P. Thakkar, C. Shah, T. Bhat and S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697412.

[8] Wikipedia contributors. (2022, February 20). Machine learning. Wikipedia. https://en.wikipedia.org/wiki/Machine_learning

[9] Zaini, N. S., Stiawan, D., Razak, M. F. A., Firdaus, A., Wan Din, W. I. S., Kasim, S., & Sutikno, T. (2020). Phishing detection system using nachine learning classifiers. Indonesian Journal of Electrical Engineering and Computer Science, 17(3), 1165. https://doi.org/10.11591/ijeecs.v17.i3.pp1165-1171

[10] Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) (3rd ed.). Morgan Kaufmann.

[11] Khanna, N. (2021, August 18). J48 Classification (C4.5 Algorithm) in a Nutshell. Medium. https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e

[12] UCI Machine Learning Repository: Phishing Websites Data Set. (2015). Rami Mustafa A Mohammad. https://archive.ics.uci.edu/ml/datasets/Phishing+Websites

[13] A. A. A. Abdulrahman, A. Yahaya, and A. Maigari, “Detection of phishing websites using Random Forest and XGBoost algorithms,” International Journal of Pure and Applied Sciences, vol. 2, no. 3, pp. 1–14, 2019.

[14] S. P. Ripa, F. Islam and M. Arifuzzaman, "The Emergence Threat of Phishing Attack and The Detection Techniques Using Machine Learning Models," 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021, pp. 1-6, doi: 10.1109/ACMI53878.2021.9528204.

[15] Al-Sarem, M., Saeed, F., Al-Mekhlafi, Z. G., Mohammed, B. A., Al-Hadhrami, T., Alshammari, M. T., Alreshidi, A., & Alshammari, T. S. (2021). An optimized stacking ensemble model for phishing websites detection.

[16] Abdelhamid, Neda. (2016). Website Phishing DataSet. Retrieved From https://archive.ics.uci.edu/ml/datasets/Website+Phishing

[17] "UCI Machine Learning Repository: Phishing Websites Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites#. [Accessed: 29-Jan-2017].

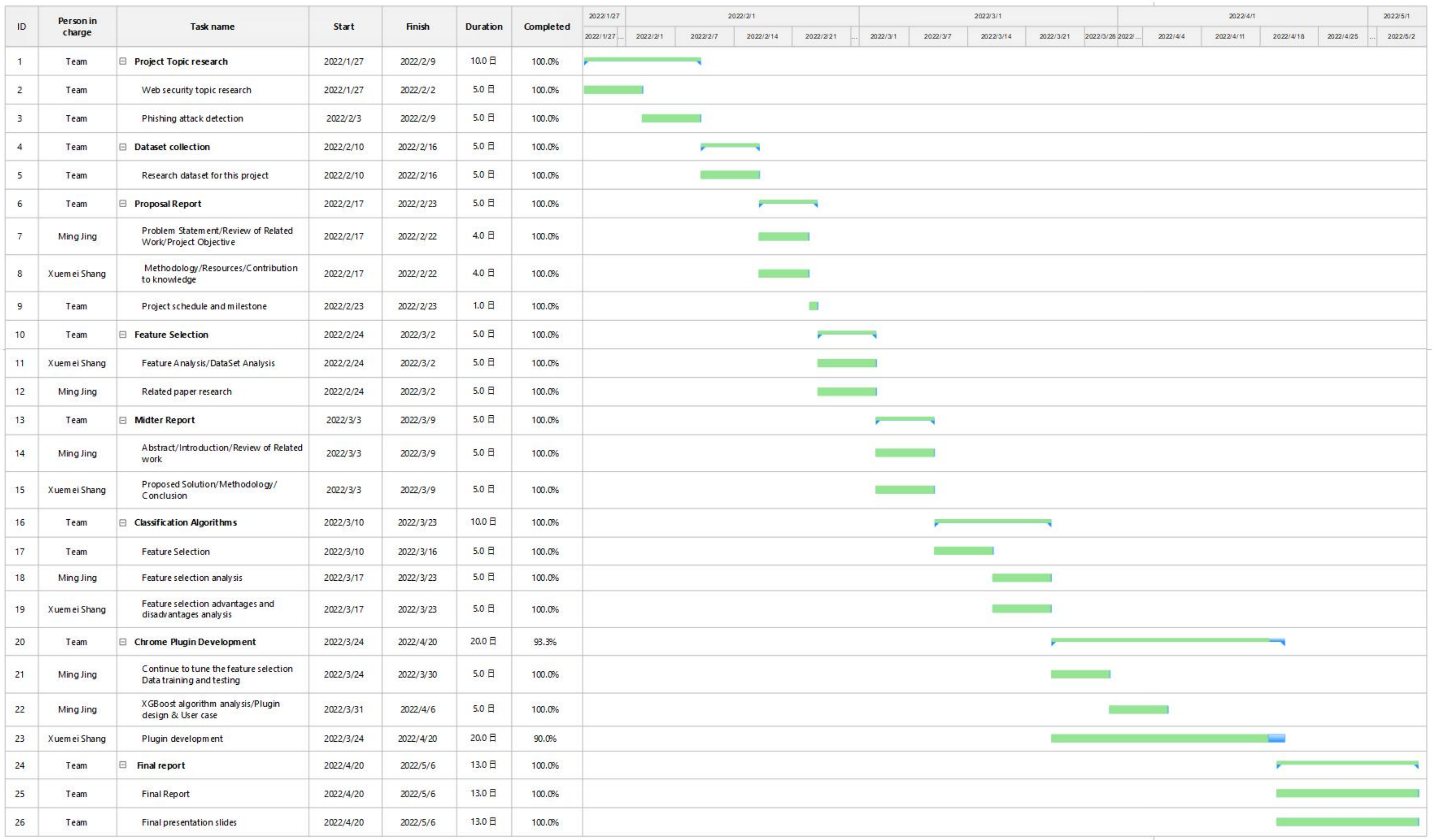
[18] Rami M. Mohammad, Fadi Thabtah & Lee McCluskey, "Phishing websites features," http://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf.

[19] Brownlee, J. (2019, November 26). How to Choose a Feature Selection Method For Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/#:~:text=Feature%20selection%20is%20the%20process

[20] Naik, R. (2022, April 4). Malicious Web Content Detection using Machine Learning. GitHub. https://github.com/philomathic-guy/Malicious-Web-Content-Detection-Using-Machine-Learning

[21] How to Create Your Own Google Chrome Extension. (2022, February 3). FreeCodeCamp.org. https://www.freecodecamp.org/news/building-chrome-extension/

Appendix: Project timeline



ID	Person in charge	Task name	Start	Finish	Duration	Completed
1	Team	⊟ **Project Topic research**	2022/1/27	2022/2/9	10.0 日	100.0%
2	Team	Web security topic research	2022/1/27	2022/2/2	5.0 日	100.0%
3	Team	Phishing attack detection	2022/2/3	2022/2/9	5.0 日	100.0%
4	Team	⊟ **Dataset collection**	2022/2/10	2022/2/16	5.0 日	100.0%
5	Team	Research dataset for this project	2022/2/10	2022/2/16	5.0 日	100.0%
6	Team	⊟ **Proposal Report**	2022/2/17	2022/2/23	5.0 日	100.0%
7	Ming Jing	Problem Statement/Review of Related Work/Project Objective	2022/2/17	2022/2/22	4.0 日	100.0%
8	Xuemei Shang	Methodology/Resources/Contribution to knowledge	2022/2/17	2022/2/22	4.0 日	100.0%
9	Team	Project schedule and milestone	2022/2/23	2022/2/23	1.0 日	100.0%
10	Team	⊟ **Feature Selection**	2022/2/24	2022/3/2	5.0 日	100.0%
11	Xuemei Shang	Feature Analysis/DataSet Analysis	2022/2/24	2022/3/2	5.0 日	100.0%
12	Ming Jing	Related paper research	2022/2/24	2022/3/2	5.0 日	100.0%
13	Team	⊟ **Midter Report**	2022/3/3	2022/3/9	5.0 日	100.0%
14	Ming Jing	Abstract/Introduction/Review of Related work	2022/3/3	2022/3/9	5.0 日	100.0%
15	Xuemei Shang	Proposed Solution/Methodology/ Conclusion	2022/3/3	2022/3/9	5.0 日	100.0%
16	Team	⊟ **Classification Algorithms**	2022/3/10	2022/3/23	10.0 日	100.0%
17	Team	Feature Selection	2022/3/10	2022/3/16	5.0 日	100.0%
18	Ming Jing	Feature selection analysis	2022/3/17	2022/3/23	5.0 日	100.0%
19	Xuemei Shang	Feature selection advantages and disadvantages analysis	2022/3/17	2022/3/23	5.0 日	100.0%
20	Team	⊟ **Chrome Plugin Development**	2022/3/24	2022/4/20	20.0 日	93.3%
21	Ming Jing	Continue to tune the feature selection Data training and testing	2022/3/24	2022/3/30	5.0 日	100.0%
22	Ming Jing	XGBoost algorithm analysis/Plugin design & User case	2022/3/31	2022/4/6	5.0 日	100.0%
23	Xuemei Shang	Plugin development	2022/3/24	2022/4/20	20.0 日	90.0%
24	Team	⊟ **Final report**	2022/4/20	2022/5/6	13.0 日	100.0%
25	Team	Final Report	2022/4/20	2022/5/6	13.0 日	100.0%
26	Team	Final presentation slides	2022/4/20	2022/5/6	13.0 日	100.0%