

Know URL: Machine Learning-based Chrome Extension for Detection of URL Phishing

Thejaswini S¹, Sinchana P², Vibha Kestur T A³, Vinya Kestur T A⁴ and Thanushree K⁵

¹⁻⁵Siddaganga Institute of Technology/ Computer Science and Engineering, Tumkur, India

Email: thejaswinis@sit.ac.in, {1si18cs109@sit.ac.in, 1si18cs129@sit.ac.in, 1si18cs132@sit.ac.in, 1si18cs125@sit.ac.in}

Abstract—Phishing is a con game that scammers/imposters use to collect personal information from unsuspecting users. It's a major problem that is spreading year after year very rapidly, especially increased by 220 phishing incidence during the pandemic period. Traditionally, phishing detection has relied on either a blacklist of known phishing URLs or a heuristic evaluation of the features of a suspected phishing website. The heuristic algorithm uses trial and error to determine the threshold for classifying harmful and benign links. The disadvantages of this method include its lack of accuracy and flexibility to new phishing URLs. To overcome the drawbacks of traditional phishing detection methods, the proposed system provides an efficient machine-learning solution for detecting phishing and fraudulent URLs. In the proposed system, model is trained using several light weight algorithms such as logistic regression, K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes and select the highly accurate and efficient model i.e., Support Vector Machine (SVM) with an accuracy of 95.7% with less load time compared to other existing models. Using the efficient model, a chrome extension phishing detection mechanism has been developed to alert the users about the illegitimate web sites and preventing the phishing attack, thereby scales down the cyber crime and cyber crisis that arises year by year.

Index Terms— Phishing Detection, Machine learning algorithms, Chrome extension, legitimate, URLs, Support Vector Machine (SVM).

I. INTRODUCTION

The availability of financial services such as online banking has made people's life simpler. One of the most significant dangers to internet security is phishing. Phishing is one of the cyber crime activity in which text messages, personally identifiable information, banking details, credit card details, password emails, telephone is been targeted. As a result, it's vital to keep such services secure and safe. It is a method of gaining user credentials through the internet by impersonating a reputable website or service. Nowadays, Phishers use Social Engineering to steal victims' personal data and the account details and it can be achieved using different techniques such as phishing, smishing, vishing, and email phishing. Phishing emails sometimes seem like emails from huge firms such as PayPal, Amazon, or Microsoft, as well as banks or government agencies. The message may instruct the victim to visit a website if consumers fall for the lure and open the link, they will be sent to a fake website. They are then prompted to provide their login and password details thus attacker receives the sign-in credentials and makes use of them to steal people's identities and sell their personal information on the black market. Phishing can be accomplished in different ways that includes via e-mail, click-baiting

advertisements, whaling, spear-phishing, and more. Attackers clone an official website into a website that seems to the user to be authentic but is really administered by the attacker. Individuals provide their credentials and other sensitive information to the cloned website, which is subsequently utilized for malevolent reasons by the attacker.

As per the Anti-Phishing Working Group (APWG) and Kaspersky Lab, phishing assaults are becoming more prevalent, posing a security risk to user information. In an attempt to address the phishing problem, several studies have lately been released. Some researchers compared the URL to a whitelist of respectable websites, while others compared it to current blacklists that include lists of harmful websites that they have been creating. The latter strategy, which makes use of a signature database, employs heuristics. In December 2021, APWG recorded 316,747 attacks, the greatest monthly number in the organization's history. Since early 2020, the number of phishing assaults has increased threefold. The difficulty of phishing detection can be overcome using machine learning techniques. So in the proposed system, phishing detection mechanism is integrated with machine learning approach. The four light weight Machine learning algorithms such as logistic regression, K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes are tested on a dataset of attributes which characterizes the behaviors commonly associated with phishing pages and based on its performance the best model is selected and chrome extension is build that classifies every browsed URL, under phished and non-phished categories on page load thereby, alerting the user of any malicious activity and preventing phishing attacks.

II. LITERATURE SURVEY

In [1][2][3][4], different features like lexical, blacklist, host based and combination of lexical and host based to detect the phishing URL. In [1], Thomas offered a study that combined a blacklisting strategy with Host-based Analysis to apply specific classifiers that might be used for detecting and removing phishing sites. The host-based, popularity-based, and lexical-based feature extractions are merged to provide a database of feature values. To extract the database's information, many machine learning algorithms are applied. Following the evaluation of the classifiers, a classifier was selected and put into use in MATLAB.

In [5], to avoid being discovered, most phishing URLs use HTTPS. There are three ways to identify phishing on the internet. The first strategy evaluates various parts of the URL, the second examines a website's authority and determines whether it is introduced or not, as well as who is in charge of it, and the third strategy confirms the website's credibility.

In [6], the paper describes CANTINA+, a comprehensive feature-based technique that combines machine learning techniques with the HTML Document Object Model (DOM), search engines, and third-party services to detect phishing. It also includes two additional filters (phish detection based on hashing near-duplicates and is a login form filter, which defines websites that do not include a login form as legitimate) to help minimize FP while also reducing runtime.

In [7], The research proposes a sophisticated approach for detecting phishing sites. The technology works as an add-on to an internet browser, alerting the user when it finds a website that is not legitimate. This proposed work is based on supervised machine learning.

In [8], The paper explores how to detect phishing websites using a predictive blacklist strategy. It used heuristics and an adequate matching algorithm to find new phishing URLs. Heuristics combined elements of known phished websites from the available blacklist to build new URLs. The URL's score is then calculated using the matching algorithm. This website is flagged as a phishing website if the score exceeds a certain threshold.

In [9] the system is designed to furnish zero hour threat protection compared with the other existing traditional techniques. In [10], SVM, Decision Tree, Random Forest, and XGBoost are used to offer a hybrid technique that incorporates the notions of bagging and boosting. This paper will serve as a reference for URL feature rules for our research and project.

III. PROPOSED SYSTEM

The proposed system can identify phishing in real time and provide an alert to the user. The main concept is to utilize machine learning algorithms on an available dataset of phishing pages to create a model that can be used to classify web pages in real time as phishing or legitimate webpage. The basic idea is to create a model into a software application that can be simply deployed to end users to protect them against phishing attacks. We opted machine learning approach to select the highest performance model and to build a Chrome extension using

JavaScript. The learnt model will be easily deployable on the Chrome Web Store, where anybody can download and utilize our phishing detection software. By using the phishing detective chrome extension, the users can have a safe browsing experience as it displays the warning message whether the browsed URL is phishing or legitimate. By displaying the warning message the proposed system protects end users against phishing attack. Selecting an efficient machine learning method for the product is mainly based on two criteria: Firstly, the trained model's accuracy should be high, as a product utilized by end customers in the actual world should not produce incorrect results. Secondly, the algorithm being used should be able to produce classifications in real time, which means it should have a short execution time and utilize little processing resources.

Considering the above mentioned two criteria the light weight machine learning algorithms such as logistic regression, K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes are used in the proposed system.

The detection of phishing mechanism is executed mainly in three phases:

- Machine Learning (ML) Analysis & Prediction Phase
- Chrome Extension Phase
- Detection Phase

The working of the three phases of proposed system is represented in the system architecture as shown in fig1.

A. ML Analysis Phase and Prediction Phase

In this phase, the dataset containing URL features are pre-processed and machine learning is applied to combat the problem of phishing attacks. Because of the vast quantity of data on phishing attack patterns, machine learning might be a suitable fit. The main concept is to use light weight machine learning algorithms on a pre-defined dataset to detect phishing in real time, analyze and visualize the dataset for better understanding, and finally train the model using several algorithms.

On compare and contrast, best model with highest accuracy is selected which will be able to produce classifications in real-time. Furthermore, the data set is processed for important features, and weights for important features are produced. These weights are passed to the detection phase.

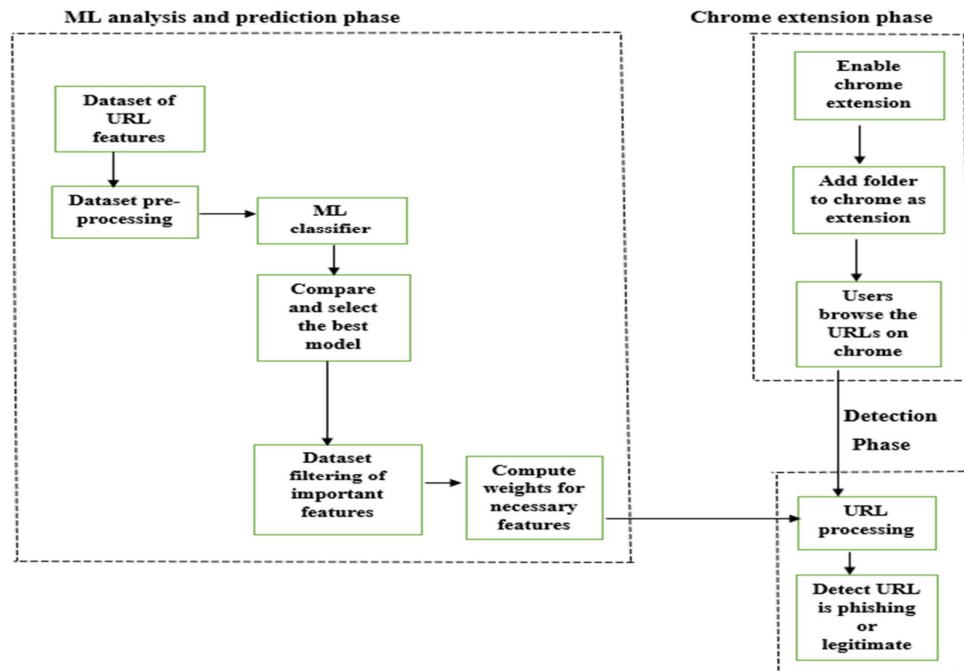


Fig.1: System Architecture

B. Chrome Extension phase

The ML model which gives the highest accuracy is used to build chrome extension. The .crx file is created after enabling the extension to our folder. The folder contains background.js, content.js, and manifest.json files which

is added to chrome as an extension. Then generate the .crx file for our folder which will be further used by the detection phase. Now the user is allowed to browse the URLs on the chrome search bar. This URL is further passed to the detection phase.

The extension is packaged to support the Chrome browser in specific, solely by the virtue of its popularity. Additionally, extensions exhibit minimal web dependence, as it collates multiple files into a single file for users to download, as a one-time activity.

C. Detection phase

In this phase, the URL processing takes place in the content.js file based on the important URL features such as URL IP address, HTTPS URL, URL portion of the anchor tag, etc... The entered URL passes through the functions of the main features and classifies every browsed URL under phished and non-phished categories.

It displays a warning message in the form of a pop-up on the user's browser screen, if the user is about to visit a phishing website by accident. To build a pop-up message, JavaScript is used and add it to the manifest file of the Chrome extension which contains the name, permissions, associated scripts, and files. This phase classifies every browsed URL, under phished and non-phished categories on page load thereby, alerting the user of any malicious activity and preventing phishing attacks.

IV. IMPLEMENTATION

Data Set :“Phishing Websites Dataset” from Kaggle is used to test and evaluate machine learning methods in the proposed system. It comprises of 11,055 URLs (instances) with 6157 phishing instances , 4898 legitimate instances and each instance consisting of 30 features/attributes as listed in Table 1. These 30 features considered in the training dataset belongs to one of the following four main category of features :

- 1) Address Bar based features
- 2) Abnormal based features
- 3) HTML and JavaScript based features
- Domain based features.

Each feature is associated with a rule. If the rule satisfies it is designated as phishing. If the rule doesn't satisfy then it is designated as legitimate. The dataset has been normalized to contain only three discrete values as shown in Fig.2:

- '1' if the rule is satisfied
- '0' if the rule is partially satisfied
- '-1' if the rule is not satisfied.

```
@relation phishing
@attribute having IP Address { -1,1 }
@attribute URL length { 1,0,-1 }
@attribute Shortening Service { 1,-1 }
@attribute having At Symbol { 1,-1 }
@attribute double slash redirecting { -1,1 }
@attribute Prefix Suffix { -1,1 }
@attribute having Sub Domain { 1,0,1 }
@attribute SSLfinal State { -1,1,0 }
@attribute Domain registration length { -1,1 }
@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,-1,0 }
@attribute SFH { -1,1,0 }
@attribute Submitting to email { -1,1 }
@attribute Abnormal URL { -1,1 }
@attribute Redirect { 0,1 }
@attribute on mouseover { 1,-1 }
@attribute RightClick { 1,-1 }
@attribute popupwindow { 1,-1 }
@attribute IFrame { 1,-1 }
@attribute age of domain { -1,1 }
@attribute web traffic { -1,0,1 }
@attribute Page Rank { -1,1 }
@attribute Google Index { 1,-1 }
@attribute Links pointing to page { 1,0,-1 }
@attribute Statistical report { -1,1 }
```

Fig.2. Dataset Attributes

When building a model for real-time deployment, there are two parameters that should be prioritized. To begin, select the dataset and train the model so that its accuracy is high so that the end user does not receive incorrect results. Second, in order to provide real-time protection, an algorithm that loads quickly on Chrome is required; that is, the algorithm must take a short amount of time to execute and does not require the user to wait long for the result, four lightweight algorithms such as logistic regression, K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes , are considered to test the system which takes short time to load on chrome. This ML model has been properly trained and an accurate model has been selected so that the end-user receives accurate results.

TABLE I: LIST OF 30 FEATURES IN THE DATASET

Address Bar Based Features:	SFH
having_IP_Address	Submitting_to_email
URL_Length	Abnormal_URL
Shortning_Service	HTML & Java Script based Features:
having_At_Symbol	Redirect
double_slash_redirecting	on_mouseover
Prefix_Suffix	RightClick
having_Sub_Domain	popUpWidnow
SSLfinal_State	Iframe
Domain_registration_length	Domain based Features:
Favicon	age_of_domain
Port	DNSRecord
HTTPS_token	web_traffic
Abnormal based Features:	Page_Rank
Request_URL	Google_Index
URL_of_Anchor	Links_pointing_to_page
Links_in_tags	Statistical_report

A. Pre-Processing Dataset

Pre-processing the URL dataset entails eliminating all null values and is a chore of cleaning the data and preparing it for the machine learning model in order to improve the model's accuracy and effectiveness. The dataset is split into a training set and a test set in the project. The performance of our machine learning model can be improved, making this one of the most important processes in the pre-processing of the data.

B. Compare & Select the best model

The training and evaluation of various ML algorithms with the dataset were carried out. The dataset is trained with different algorithms and tested the accuracy score of each algorithm and chose the best one. The dataset is split into 8:2 ratios, (80% for training and 20% for testing). In this system, accuracy, f1 score, recall, and precision metrics were used to compare and contrast the four different lightweight algorithms as shown in fig 3.

	ML Model	Accuracy	f1_score	Recall
0	Support Vector Machine	0.957	0.963	0.982
1	K-Nearest Neighbors	0.953	0.959	0.990
2	Logistic Regression	0.924	0.933	0.947
3	Naive Bayes Classifier	0.583	0.420	0.291

Fig.3. Model Comparison

Since Support Vector Machine got the highest accuracy score, it was chosen as the best and the final model.

C. Feature Selection

The data set contains 30 features, some of which are irrelevant and may degrade the model's performance. As a result, important features must be chosen. To select the strong features, two techniques are used: feature selection by correlation matrix with heatmap (as shown in fig 4) and feature importance. Dataset filtering is done based on feature weights to remove unnecessary features that will degrade model performance.

Weight calculation:

The feature that gives zero weight is removed, while the remaining features that affect the model are kept .

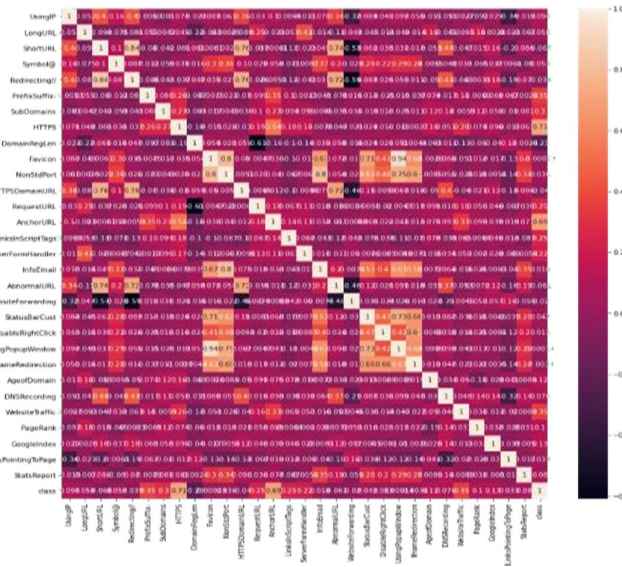


Fig. 4. Correlation matrix with heatmap

SVM produces a hyperplane that effectively distinguishes the classes. By returning the coordinates of a vector orthogonal to the hyperplane, the coefficients returned by `SVM.coef_` serve as the weights that represent this hyperplane. Consider the vector w . We can determine which side it is by taking the dot product of any point with the vector because its direction provides us with the expected class. If the dot product is positive, it belongs to the positive class; if it is negative, it belongs to the negative class. Therefore, among 30 features (Table 1) only the important 16 features are selected and are listed in the Table 2.

having_IP_Address	HTTPS_token
URL_Length	Request_URL
Shortening_Service	URL_of_Anchor
having_At_Symbol	Links_in_tags
double_slash_redirecting	SFH
Prefix_Suffix	Submitting_to_email
having_Sub_Domain	on_mouseover
Favicon	Iframe

D. Chrome extension phase

Chrome extensions are programs that can be installed in Chrome to alter its functionality. By merging current web browser functions, the extension can give additional functionality and allow users to perform multiple tasks at once. The Google Chrome Extension is based on and installed inside the Chrome browser, and it has a small impact on the computer system's resource usage. Chrome is a widely used browser that is simple to use. The browser extension will be minimal and simple to install. Because it only takes one click to install, the Google Chrome extension is both safe and user-friendly. Users can disable the extension with a single click when they want to.

The first step in creating the Chrome extension is to generate the "manifest.json" file in the extension folder. This JSON file provides the extension's name as well as its version. It also contains URLs to the popup HTML file and the content script. The content.js file runs in a separate, unprivileged JavaScript environment with full DOM access. The high-accuracy trained model will be utilized as a persistent model to classify websites in this case.

To create a .crx file, open Google Chrome and click on the Chrome menu, go to more tools, select the extension option and enable the developer mode. Then, click the "Pack Extension" button to upload the folder from our computer which will generate the .crx file for our folder which will be further used by the detection phase. If we need access to external extensions or APIs, background.js is required to establish a

communication channel between content.js and privileged components of our extension, a process known as message passing.

The steps to add the extension:

1. Open Google Chrome on your computer.
2. Click on : dots.
3. Go to more tools, and select extensions.
4. Switch on the developer mode.
5. Click load unpacked.
6. Browse the folder that you want to add as an extension.
7. Click on add, then the extension is added successfully and ready to use.

E. Detection phase

The Phishing Detection Chrome Extension aims to classify, every browsed URL, under phished and non-phished categories (on page load) thereby, alerting the user of any malicious activity and preventing phishing attacks. With this development, phishing websites may be detected before the page even loaded, alerting visitors before they input any sensitive information to the phishing website. To build a pop-up message, use JavaScript and add it to the manifest file of the Chrome extension which contains the name, permissions, associated scripts, and files.

The user enters a URL, the URL processing takes place and in the content.js file, we will check mainly the URL features such as URL IP address, HTTPS URL, In [8] URL portion of the anchor tag, etc... The entered URL passes through the functions of the main features and classifies every browsed URL under phished and non-phished categories.

V. EXPERIMENTAL RESULTS

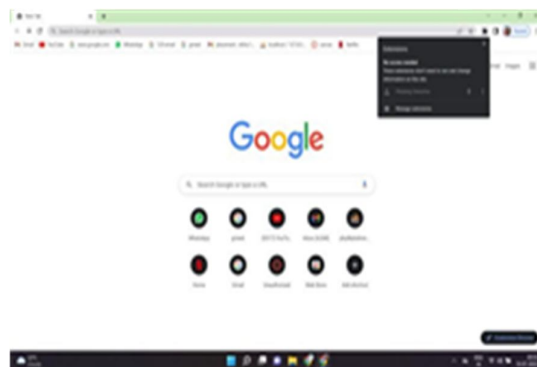


Fig.5. google chrome extension added successfully

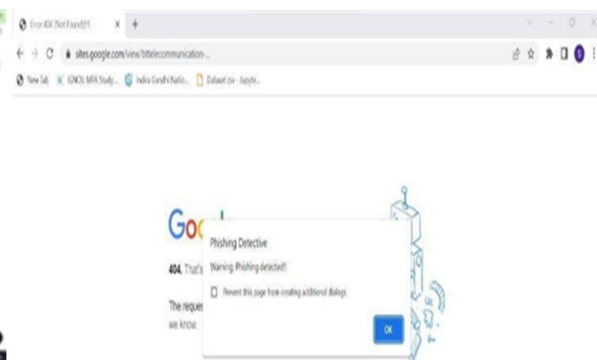


Fig.6. Detection of phishing URL with pop-up message

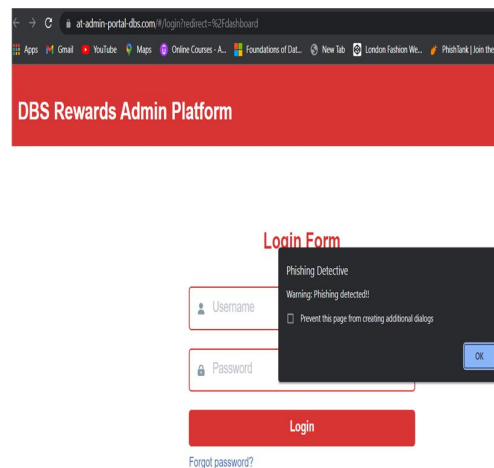


Fig.7. Magnified image of phishing URL with a pop-up message

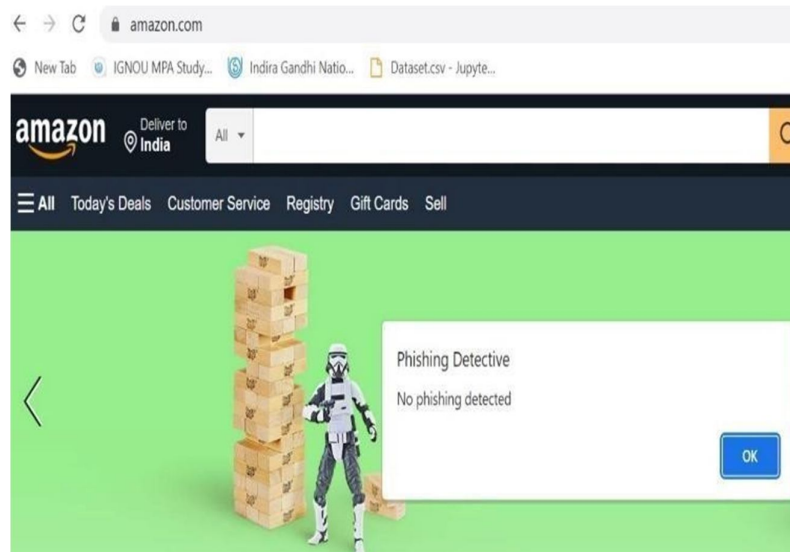


Fig.8. Magnified image of legitimate URL with a pop-up message

The above Fig.5, Fig.6, Fig.7, and Fig.8 shows the final experimental results of the proposed approach. By selecting the minimum number of features, the system has achieved the highest accuracy of 95.7% using the lightweight models which takes low loading time.

VI. CONCLUSION

Phishing poses a serious danger to internet security, and safety, and it is crucial to protect internet users from falling prey to online fraud. Users need phishing detection technologies to have a safe online experience. Phishing detection techniques that are traditional in nature are ineffective. The limitations of conventional methods are overcome by the machine learning approach. Various models are trained using features such as the presence of @ in the URL and double slash redirection, among others. Detecting phishing is difficult due to the use of URL obfuscation to shorten the URL, link redirections, and modifying links to make them appear trustworthy. Many researchers have been motivated to work on several techniques to improve phishing attack detection accuracy and lower false alarm rates.

As a result, a Chrome extension is developed that uses a machine learning algorithm to automate the task of increasing the accuracy of detection and overcoming drawbacks of old approaches. After examining several algorithms, Support Vector Machine (SVM) with the highest accuracy of 95.7% was selected as the core algorithm. The extension makes it simple for end users to employ our phishing detection technology which notifies the user by pop-up message. For future enhancements, the performance factors, such as speed and simplicity, can be upgraded to provide users with a better experience, and the Chrome extension can be uploaded to the Chrome web store for public usage.

REFERENCES

- [1] Joby James, Sandhya L, Ciza Thomas, "Detection of Phishing Website Using Machine Learning Approach", 2023 International Conference on Control Communication and Computing (ICCC).
- [2] Huaping Yuan, Xu Chen, Yukun Li, Zhenguo Yang, Wenyin Liu, "Detecting Phishing Websites and Targets Based on URLs and Webpage Links", 2018 24th International Conference on Pattern Recognition (ICPR) <https://ieeexplore.ieee.org/document/8546262>
- [3] Jitendra Kumar; A. Santhanavijayan; B. Janet; Balaji Rajendran; B.S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning", 2020 International Conference on Computer Communication and Informatics (ICCCI) <https://www.hindawi.com/journals/scn/2017/5421046/>
- [4] Mohammed Nazim Feroz, Susan Mengel, "Phishing URL detection using URL Ranking", 2015 IEEE International Congress on Big Data, BigData Congress 2015. <https://scholars.ttu.edu/en/publications/phishing-url-detection-usingurl-ranking-5>

- [5] Mahajan Mayuri Vilas; Kakade Prachi Ghansham; Sawant Purva Jaypralash, "Detection of Phishing Website Using Machine Learning Approach", 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECOT) <https://ieeexplore.ieee.org/abstract/document/9114695>.
- [6] Guang Xiang, Jason Hong, Carolyn P. Rose, LorrieCranor, "Cantina+: A feature-rich machine learning framework for detecting phishing websites", 01 September 2011 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.403.5479>
- [7] Amani Alswailem; Bashayr Abdullah; Norah Alrumayh; Aram Alsedrani, "Detecting Phishing Websites Using Machine Learning", 2019 2nd International Conference on Computer Applications Information Security (ICCAIS) <https://ieeexplore.ieee.org/document/8769571>
- [8] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks", 2010 Proceedings IEEE INFOCOM <https://ieeexplore.ieee.org/document/5462216>
- [9] Aaron Blum, Brad Wardman, Tamar Solorio, GaryWarner, "Lexical Feature Based Phishing URL Detection Using Online Learning",Conference Proceedings of the 3rd ACM Workshop on Security and Artificial Intelligence, AISec 2010, Chicago, Illinois, USA, October 8, 2010 https://www.researchgate.net/publication/221609867_Lexical_feature_based_phishi
- [10] Nusrath Tabassum; Farhin Faiza Neha; Md. Shohrab Hossain; Husnu S. Narman,"Detection Technique through Dimensionality Reduction", 2021 IEEE International Black Sea Conference Communications and Networking (BlackSeaCom) <https://ieeexplore.ieee.org/document/9527806>