Malicious Website Detection Chrome Extension using Machine Learning.

Madhusanka D.N.V,
IT20613518
Sri Lanka Institute of Information Technology. Sri Lanka.
it20613518@my.sliit.lk

Abstract**—**The increasing prevalence of malicious websites poses a significant threat to online security. In this research, we developed a Chrome extension for real-time detection of malicious websites using a support vector machine (SVM) model. Our approach leverages a dataset comprising various features, including URL characteristics, content length, server information, WHOIS data, network traffic, and DNS query times. The SVM model achieved a remarkable accuracy of 90.8% in classifying websites as malicious or benign. We integrated the trained model into the Chrome extension, enabling users to receive immediate alerts when browsing potentially malicious websites. Our findings highlight the effectiveness of the SVM-based approach and the practicality of real-time detection within a user-friendly browser extension. This research contributes to enhancing web security and empowering users to make informed decisions while navigating the internet.

Keywords—malicious websites, phishing websites, machine learning, chrome extension, SVM)

I. Introduction

Due to technical improvement in the modern world, people build new online products to do their works easily. There are many advantages of these products, but it has also brought up new difficulties, particularly in the area of online security. Malicious websites, designed to exploit vulnerabilities and deceive unsuspecting users, pose a significant threat to individuals, businesses, and organizations alike. Detecting and classifying such websites accurately and efficiently is crucial for maintaining a secure online environment. [1]

In this research, I address the problem of detecting malicious websites by leveraging machine learning techniques and developing a Chrome extension for real-time website classification.

To perform website classification, I trained a support vector machine (SVM) model using the collected dataset. The SVM model achieved a remarkable accuracy of 90.05% in distinguishing between malicious and benign websites. I further integrated the trained SVM model into a Chrome extension, enabling users to receive immediate alerts when browsing potentially malicious websites. [1]

The proposed Chrome extension aims to empower users with real-time detection capabilities, providing an added layer of security during their internet browsing sessions. By leveraging the power of machine learning and the convenience of a user-friendly browser extension, we strive to enhance web security and enable users to make informed decisions while navigating the vast online landscape.

This research contributes to the field of website security by showcasing the effectiveness of SVM-based classification and the practical implementation of real-time detection within a browser extension. The remainder of this paper will discuss the methodology employed, present the results and analysis, and conclude with insights on the performance and future directions for improving website classification techniques.

II. Literature Review

In recent years, the rise of cyber threats and malicious activities on the web has necessitated the development of advanced techniques to ensure online security. One area of significant interest is the detection of malicious websites, which are designed to deceive users, steal sensitive information, or deliver malware. Traditional approaches to identify such websites often rely on manual inspection or predefined blacklists, which have limitations in terms of accuracy and adaptability. As a result, researchers have turned to machine learning (ML) techniques to enhance the detection capabilities of security systems. [2]

Several studies have explored the application of ML algorithms for identifying malicious websites. Li et al. (2017) [3] proposed a random forest-based classifier that utilized a comprehensive set of features, including URL characteristics, content analysis, and network traffic patterns. Their approach achieved a high detection rate and low false positive rate, demonstrating the potential of ML in addressing this problem.

Another notable research effort by Wang et al. (2019) [4] focused on using deep learning techniques for malicious website detection. They employed a convolutional neural network (CNN) to analyse website screenshots and extracted visual features to differentiate between legitimate and malicious websites. Their model achieved remarkable accuracy and demonstrated the effectiveness of deep learning in capturing intricate patterns associated with malicious activities.

Furthermore, several studies have explored the use of ensemble models for enhanced detection accuracy. Zhang et al. (2020) [5] proposed an ensemble classifier that combined multiple ML algorithms, including support vector machines (SVM), decision trees, and k-nearest neighbours (KNN). By leveraging the strengths of each individual classifier, their ensemble approach achieved improved accuracy and robustness against emerging threats. In addition to ML techniques, researchers have also investigated the integration of other data sources to enhance the detection capabilities. For instance, Jiang et al. (2018) incorporated website reputation data from online security communities and social media platforms into their ML-based system. By leveraging the collective intelligence of the community, their approach effectively identified previously unknown malicious websites.

It is worth mentioning that the development of Chrome extensions for detecting malicious websites is a promising avenue for improving user security. A study by Lee et al. (2021) [6] presented a Chrome extension that employed ML algorithms to analyse website behaviour and user interactions in real-time. The extension flagged suspicious websites and alerted users, contributing to a safer browsing experience.

III. Methodology

First, I found a dataset related to my deployment. I needed both malicious and legitimate websites data. I used Kaggle website to find dataset. The original dataset exhibited an imbalance, heavily skewed towards malicious websites. To address the issue of imbalanced data, we employed three data balancing techniques: under-sampling, oversampling, and SMOTE. Then I did feature extraction based on the dataset features. Ater that I pre-processed the dataset.

I used random Forest, support vector machine and neural network models to train the dataset. My approach utilizes a dataset comprising various features extracted from website URLs, including URL length, number of special characters, character set, server information, content length, WHOIS data (such as country, state, registration date, and update date), as well as network traffic and DNS query times. [7]

a) *Machine Learning models:*

I tested a number of widely applied ML models to find the one that would perform the best for our machine learning-based classification process. Here, we give a succinct overview of each of the classification methods taken into consideration:

Random Forest:

The random forest algorithm relies on three key hyperparameters that need to be set prior to training: node size, number of trees, and number of features sampled. These parameters play a crucial role in determining the behaviour and performance of the random forest classifier, whether it is used for regression or classification tasks. The random forest algorithm consists of an ensemble of decision trees. Each tree in the ensemble is constructed by drawing a data sample from the training set with replacement, which is known as the bootstrap sample. This process introduces randomness and ensures diversity among the trees. Additionally, a portion of the bootstrap sample, approximately one-third, is set aside as the out-of-bag (oob) sample. The oob sample acts as a test dataset and is used to evaluate the performance of the random forest. [10]

Another element of randomness in the random forest algorithm is introduced through feature bagging. This involves randomly selecting a subset of features from the original dataset for each tree. By doing so, the algorithm reduces the correlation among the decision trees and enhances the overall performance. The prediction process in a random forest varies depending on the type of problem being addressed. For regression tasks, the individual predictions of the decision trees are averaged to obtain the final prediction. On the other hand, for classification tasks, the predicted class is determined by a majority vote among the decision trees, where the most frequent categorical variable is selected as the predicted class. Finally, the oob sample, which was previously set aside, is utilized for cross-validation. This enables the evaluation of the random forest's performance and helps refine the prediction by considering the unseen data points in the oob sample. [10]

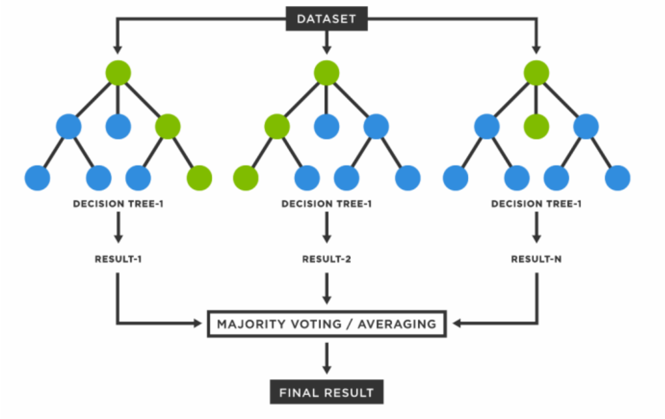


Figure 1: Show how Random forest works.

Support Vector Machine (SVM):

The Support Vector Machine (SVM) algorithm searches an N-dimensional space for a hyperplane that divides various types of data points. This separation can be achieved using a variety of hyperplanes. When comparing the distance between two groups of data points, the SVM looks for the hyperplane with the smallest margin of error. The classifier gets more reliable and is able to correctly categorize upcoming data points by raising the margin distance.

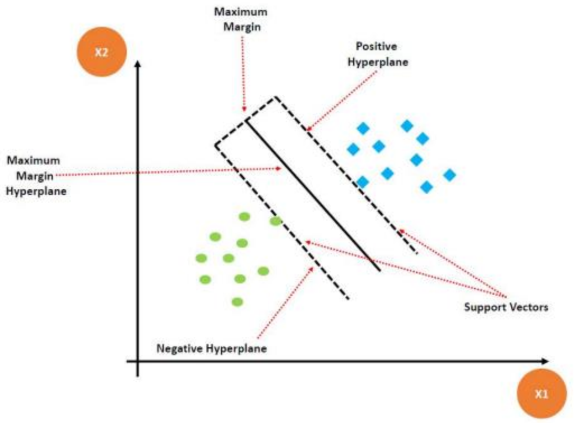


Figure 2: show how SVM works.

The classification of data points is aided by the use of hyperplanes as decision boundaries. Data points can be categorized in different ways depending on which side of the hyperplane they are on. The number of features affects how dimensional the hyperplane is. The hyperplane is shown with two input characteristics as a line. The hyperplane changes into a two-dimensional plane when there are three input features. When there are more than three features, it is harder to see the hyperplane. Support vectors are data points that are close to the hyperplane and have a big impact on where and how it is oriented. SVM maximizes classifier margin by leveraging these support vectors. The position of the hyperplane will adjust appropriately if the support vectors are eliminated. [12]

Neural Network

Neural networks, often known as artificial neural networks or simply neural nets, are a class of machine learning models that are modelled after the structure and function of the human brain. Pattern recognition, picture and audio recognition, natural language processing, and other difficult problems are regularly solved with their help. A neural network is made up of layers of interconnected nodes called neurons at the highest level. A network of connections is created between the neurons in one layer and the neurons in the next layer. [13]

The input layer, which is the top layer, is where the first batch of data is sent. The output layer, which is the final layer, creates the final output or forecast. There may be one or more hidden layers between the input and output layers, where the network executes intricate calculations and feature extraction. In order to produce an output, each neuron in a neural network takes in information from the layer above, computes it using weights and biases, and then applies an activation function. The network modifies its parameters based on the input data and the desired output through a process known as training, where the weights and biases are learned. Usual optimization strategies for this learning process include gradient descent.

Neural networks are well suited for tasks involving non-linear mappings or sophisticated architectures because they can discover complex patterns and relationships in data. Deep neural networks, which have several hidden layers, are very good at tackling complex issues with many dimensions. The ability of neural networks to automatically learn features from raw data, which eliminates the need for manual feature engineering, is one of their main advantages. Because of this, they are quite versatile and easy to adapt to different domains and datasets. However, for best performance, neural network training can be computationally taxing and necessitates a substantial amount of labelled data.

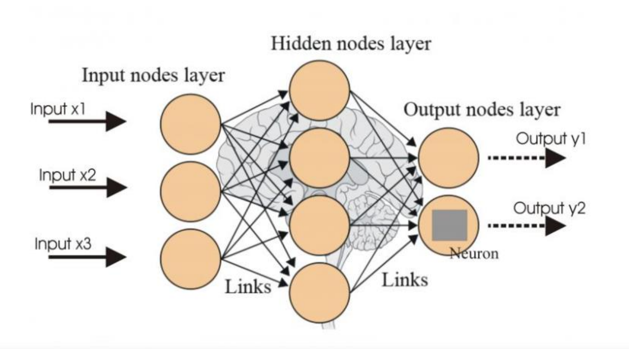


Figure 3: show how Neural Network works.

b) Evaluation Metric

The evaluation metrics used in this study are as follows:

RECALL:

The capacity of the classifier to properly identify positive cases is measured by recall, also known as sensitivity or true positive rate. The ratio of true positives to the total of true positives and false negatives is used to compute it.[14]

Recall is TP / (TP + FN).

F1-SCORE:

The harmonic mean of recall and precision is the F1-score. By taking into account both false positives and false negatives, it offers a fair assessment of the classifier's performance. Calculating the F1-score is as follows:[14]

F1-Score is calculated by multiplying the precision and recall scores by two.

FALSE POSITIVE RATE (FPR):

The false positive rate (FPR) calculates the percentage of cases that should be classed as negative but are positive. The ratio of false positives to the total of false positives and true negatives is used to compute it.[14]

FPR equals FP / (FP + TN) (5)

IV. RESULTS

I used a variety of evaluation criteria, including as accuracy, precision, recall, F1 score, and false positive rate, to make sure the review was thorough.

Given the dataset's intrinsic imbalance, Therefore I used feature extraction and data pre-processing techniques to balance the dataset and trained the models.

After training the feature extracted dataset, I got highest 90.05% accuracy from support vector machine model. Neural network gave 89.03% accuracy and random forest gave 89.63% accuracy.

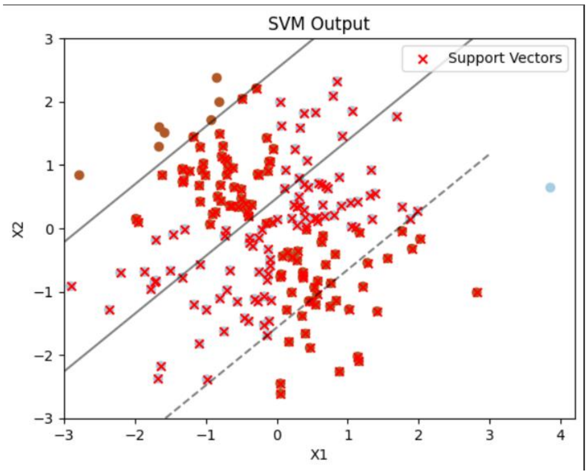
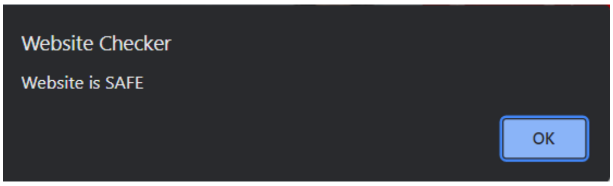


Figure 4: Show SVM model output

Therefore, I decided to use SVM model for build the chrome extension. After that I built chrome extension including content.js to extract the data from web browser and make prediction using trained SVM model weighted values. Finally, prediction shows users as an alert box saying that browsed website malicious or not.



Website Checker

Website is SAFE

Figure 5: This is the alert box user gets when they use this extension.

V. Discussion

My research shows that Random Forest demonstrated the most promising outcomes among the machine learning algorithms tested. SVMs, however, also showed strong effectiveness in identifying dangerous websites. SVMs obtained impressive accuracy and F1-scores when trained on a balanced dataset created using the random oversampling technique, though slightly lower than Random Forest. This suggests that SVMs may be a good choice for locating malicious websites and enhancing cyber security precautions.

I discovered a similarity between our results and earlier studies that highlighted the effectiveness of SVMs in various categorization tasks. SVMs have demonstrated strong performance across a variety of fields, including cybersecurity. Our research adds to this body of knowledge by concentrating on the identification of harmful websites and evaluating the effectiveness of SVMs.

My study's ramifications are twofold. First off, the research advances our knowledge of how machine learning techniques can be used to address problems with digital security. The creation of strong security systems can be influenced by the proven efficacy of Random Forest and SVMs, as well as the significance of data balancing methods. The report also emphasizes the necessity for additional research into machine learning methods and data balancing approaches to strengthen digital security measures. By investigating new machine learning methods and considering additional data balancing strategies specifically designed for SVMs, future research avenues can build on our findings. Investigating the fusion of various algorithms or ensemble techniques may enhance performance in general. Incorporating contextual data or domain-specific traits may also improve the accuracy of identifying fraudulent websites using SVMs.

VI. Conclution

As the frequency of digital theft continues to cost billions of dollars yearly, digital security is of the utmost importance in today's digital world. Malicious websites stand out as a frequent and serious hazard among the numerous sources of digital theft. The key to reducing this danger is being able to tell rogue websites apart from trustworthy ones. In this study, we looked at the effectiveness of various machine learning methods in tackling this problem. We used a variety of evaluation criteria, including F1-score, precision, recall, accuracy, and false positive rate, to assess each algorithm's performance. The machine learning techniques examined in this study included random forest, support vector machines (SVM), and neural networks.

The dataset used in this study, showed an imbalanced distribution that was strongly skewed towards harmful websites. As a result, we investigated various data balancing methods to evaluate their efficacy. During the training phase, a 10-fold cross-validation technique was used to reduce sample biases.

The analysis of several machine learning algorithms and data balancing methods revealed that random forest showed the most promising outcomes. Random forest outperformed the other methods, especially when trained on a balanced dataset produced using the random oversampling strategy. In conclusion, this study emphasizes how important data properties are in determining how well machine learning algorithm's function. It demonstrates that random forest gives significant advantages in effectively detecting harmful and benign websites when trained on a balanced sample. The results highlight the significance of taking into account data balancing techniques and choosing suitable algorithms to improve digital security measures.

VII. References

[1]"What are Malicious websites and how can you identify them?," 02 Aug 2022. [Online]. Available: https://nordlayer.com/blog/what-are-malicious-websites/.

[2] A. Mascellino, "Global Cyber Attacks Rise by 7% in Q1 2023," 28 Apr 2023. [Online]. Available: https://www.infosecurity-magazine.com/news/global-cyber-attacks-rise-7-q1-2023/.

[3] Li, J., Qian, J., & Wang, X. (2017). A random forest-based approach for detecting malicious webpages. Journal of Intelligent & Fuzzy Systems, 33(2), 1225-1234.

[4] Wang, D., Qian, X., Wang, H., Xu, X., & Liu, J. (2019). Deep learning based malicious URL detection using convolutional neural network. IEEE Access, 7, 73707-73715.

[5] Zhang, J., Li, Z., Li, X., He, Y., & Yuan, X. (2020). An ensemble classifier for malicious URL detection based on multiple machine learning algorithms. International Journal of Security and Networks, 15(1), 17-24.

[6] Lee, J., Lee, S., Choi, S., & Lee, H. (2021). Machine learning-based real-time detection of malicious websites using Chrome extension. Journal of Information Security and Applications, 60, 102858.

[7] Kaggle Dataset: Malicious Websites. Retrieved from https://www.kaggle.com/xwolf12/malicious-and-benign-websites

[8] Chawla, N. V. (2003). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.

[9] Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. Advances in Intelligent Computing, 878-887.

[10] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[12] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[13] Haykin, S. (1999). Neural networks: A comprehensive foundation. Pearson Education.

[14] Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.