Malicious URL Detection

Shakti Kinger

Dr. Vishwananth Karad MIT-WPU, S.No.124, Paud Road, Kothrud, Pune, 411038, Maharashtra, India

Email:shakti.kinger@mitwpu.edu.in

Aniket Sharma

Dr. Vishwananth Karad MIT-WPU, S.No.124, Paud Road, Kothrud, Pune, 411038, Maharashtra, India

Email:aniketsharma2468@gmail.com

Pranav Nirmal

Dr. Vishwananth Karad MIT-WPU, S.No.124, Paud Road, Kothrud, Pune, 411038, Maharashtra, India

Email:nirmalpranav187@gmail.com

Ayush Shrivastav

Dr. Vishwananth Karad MIT-WPU, S.No.124, Paud Road, Kothrud, Pune, 411038, Maharashtra, India

Email:ayushnshrivastav@gmail.com

Sakshi Saindane

Dr. Vishwananth Karad MIT-WPU, S.No.124, Paud Road, Kothrud, Pune, 411038, Maharashtra, India

Email:sakshi.saindane15@gmail.com

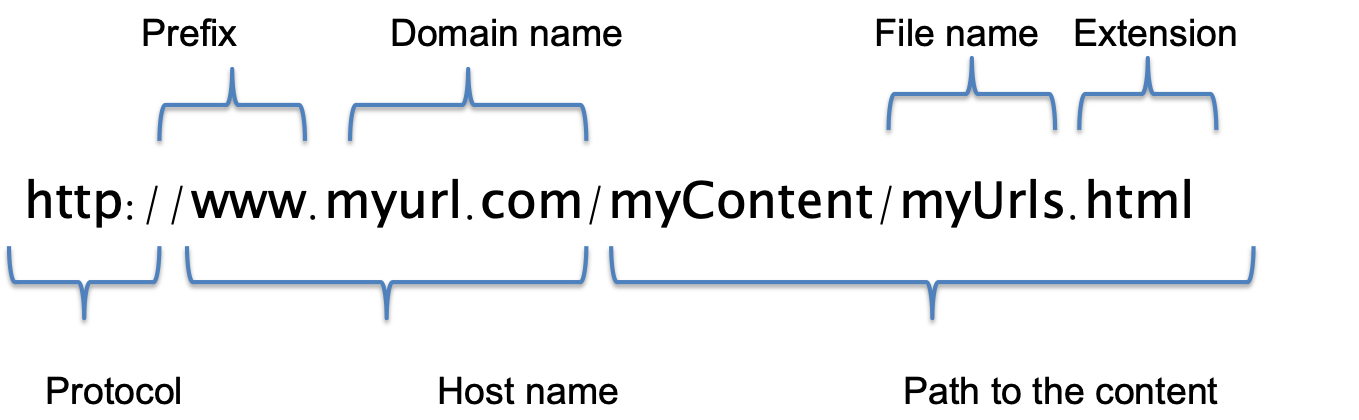
***Abstract-*** **Malicious is a term used to describe a desire to harm. A dangerous website seeks to damage the user by propagating malware, infecting the victim's system, and gathering crucial information. As a result of the worldwide lockdown in 2020, the use of internet services to do business while staying at home has altered and risen. Cybercriminals committed an increasing number of cybercrimes as a result, and businesses suffered large data losses. To stop these assaults, malicious URLs must be located and threat kinds must be recognised. Static features characterising these behaviours may be used to characterise how rogue web sites import vulnerabilities from distant domains and hide exploit code in order to identify the vast majority of malicious web pages.**

***Keywords - Malicious URL detection; feature extraction; feature selection; machine learning***

1. INTRODUCTION

The pace of technological innovation is accelerating in the modern world. The Internet has had a comparable evolution to that of technology. The use of the internet for social and commercial purposes is rapidly expanding. The prevalence of online use for these objectives broadens the potential for cybercrime. Attackers rise in number in direct proportion to connection and user growth. Victims include the government, business, and people. Predicting future dangers and their nature is a tough and, in many cases, intractable undertaking. One of the main risks to cyber security is malware or malicious websites. While malicious URLs in particular pose a significant danger to online security. A frequent and significant risk to cyber security is a malicious URL. Malicious URLs host content that is anomalous, such as spam, phishing attempts, user exploitation, etc. They let unauthorised users be attacked by cars. Every year, they experience enormous financial losses of billions of dollars. First and foremost, it is crucial to regularly identify and respond to such assaults for security [2]. Usually, large blacklists are used to perform such detections [2]. Exhaustive blacklists are impractical in real life [2]. The billions of URLs that people encounter on a daily basis cannot be addressed by the crude detection algorithms now in use. Large-scale binary classification problems are addressed using machine learning techniques [8]. The classification of good and bad URLs can be done with great accuracy using a variety of classifiers in machine learning [1][2][8]. Additionally, Huang et al. [10] uses a greedy selection approach to identify malicious URLs. The experimental investigation on URL detection using machine learning algorithms by Liu et al. is similar [11]. Using a Decision Tree approach, Vu et al. conducts cost-sensitive malicious URL detection [12]. In this empirical study, we carry out the following tasks: (a) gathering a dataset made up of a large number of URLs, including both malicious and non-malicious URLs; (b) dividing the collected dataset into two subsets in the ratio of 80:20 for training purposes and testing purposes; (c) extracting features from the training data classified into lexical features, network-based features, and host-based features; and (d) training the system using the training data and machine learning algorithm In order to detect malicious URLs, and want to provide a thorough examination using machine learning techniques like the Decision Tree, Random Forest, Autoencoder, XGBoost, Multilayer perceptrons and SVM algorithms.

1. RELATED WORK

To identify rogue websites, a number of researchers have suggested using machine learning, data mining, and even deep learning approaches. The quality and mix of key characteristic features of web pages, such as network traffic data, content traits, lexical characteristics of URLs, and even domain name system (DNS) data, are crucial to the effectiveness of these strategies [8]. 

However, obtaining these attributes can be expensive and occasionally calls for downloading entire web pages [9] or searching through numerous DNS servers and ISPs to find enrichment data like geo-location, registration records, and network information [6], which face the issue of network latency and are therefore unsuitable for real-time systems [7][6]. Previous works have explored the use of URL lexical features only, and it has been proven that URL features alone can produce an accurate means of detecting malicious webpage in real-time systems [7][9][11] despite the challenges associated with using non-lexical features to detect malicious URLs in real-time despite their high accuracy values [6][9].

The performance reported throughout the literature, even in systems that solely use lexical-based features, is also greatly influenced by the model and dataset used [7][9]. For instance, in [11], the authors examined two unsupervised machine learning models and three supervised machine learning models (k-nearest neighbour, support vector machine (SVM), and naïve bayes classifier) (k-means and affinity propagation). The analysis's findings showed that supervised models slightly outperformed unsupervised machine learning models. The amount of the dataset itself could also be problematic. For instance, in [4], association rule mining and a number of machine learning models were combined to categorise URLs as harmful or benign based on URL-derived attributes. The Synthetic Minority Over-Sampling Technique was used by the authors to address the class imbalance and small dataset size (SMOTE). The comparison of the models' performance before and after class balancing demonstrates that most models significantly improved after class balancing and the ensuing increase in dataset size.

While many researchers have used well-known machine learning techniques to categorise URLs as dangerous, some of these studies have compared ML models on their dataset, there is still a significant issue because these analyses are typically dataset-specific. In order to avoid producing erroneous results that are caused by the dataset utilised rather than the analysis conducted, as demonstrated in several earlier research efforts in other domains, it is crucial for this research to understand how machine learning models generalise over a variety of datasets. The main contribution of this work will be to aid academic researchers and security specialists in understanding what functions best under various conditions without sacrificing critical characteristics of malicious website detection systems like speed, high accuracy, low false-negative rates, and so on.

1. PROPOSED SYSTEM

For classification purposes, the proposed system employs Machine Learning algorithms and analyses using the various features received from the URL. The full flow diagram of our suggested system, shown in Figure 1, includes the phases of data collection, feature extraction, data preprocessing, model training, model testing, model selection, and final output phase. It also represents the point of origin for data collection, the features that were extracted, the models that were applied for classification, and the output, which consists of precision and classification outcome.

Data Gathering trustworthy and illuminating datasets is a crucial step in solving learning-based classification or regression problems. For training and testing purposes, data with labels that include both harmful and non-malicious URLs must be gathered from a reputable source in order to improve classification accuracy.

Data Collection: Preparing the data includes: Randomly selecting URLs: The dataset from http://data.phishtank.com/data/online-valid.csv, which contained hundreds of phishing URLs, was taken into consideration. To reduce the possibility of data imbalance, a margin value of 10,000 phishing URLs and 5000 real URLs was taken into account. We chose 5000 samples at random from the dataframe.

Feature Selection and Extraction: When dealing with a large dataset, the phase of feature selection and extraction is crucial and challenging. A feature in machine learning is a distinct measurable quality, trait, or attribute of a phenomenon being observed. For effective algorithms, selecting informative, differentiating, and independent features is a crucial step.

Tokenizing the URLs: The dataset's "URLs" column was tokenized. A total of 5000 reliable URLs were stored and utilised to extract model training features.

Feature Extraction: The URLs were processed using the following functions to extract features:

Domain Extraction: This feature extraction function pulls out the URL's domain. It provides details about the URL's origin.

Check for IP Address (Have IP): This function determines whether the URL has an IP address. If the URL points directly to an IP address rather than a domain name, this can help determine that.

· Check for the "@" symbol (Have At): This function determines whether the URL contains the "@" symbol. The '@' mark in a URL can be a red flag that it isn't a standard URL and should raise suspicion.

URL Length Categorization (URL Length): This function calculates the URL's length and classifies it. The intricacy and structure of the URL can be inferred from the URL's length.

Number of slashes in the URL (URL Depth): This method returns the quantity of slashes in the URL. The amount of slashes in the URL might provide insight into its structure and depth.

Check for Redirection '//' (Redirection): This function determines whether the URL contains the redirection character. Redirections in URLs can mean that a user is being sent to a different website, which raises several warning flags.

The function that determines whether the "HTTPS" token is present in the domain portion of the URL is called "Presence of HTTPS Token in Domain Part of URL" (https Domain). Although the existence of "HTTPS" can suggest that a website is safe, it does not ensure the website's reliability.

This function, Check for Shortening Services (Tiny URL), determines whether the URL is being shortened. The original URL may be hidden when using a shortening service, which makes it more challenging to verify the legitimacy of a website.

Prefix/Suffix Separation in Domain (Prefix/Suffix): This function determines whether the domain name contains a prefix or suffix that is divided by a hyphen (-). This may be a sign that the domain name was made with an ulterior motive and is not genuine.

Web Traffic (Web Traffic): This feature offers data on the volume of traffic that a website receives. A website's high traffic volume may be a sign that it is a trustworthy website.

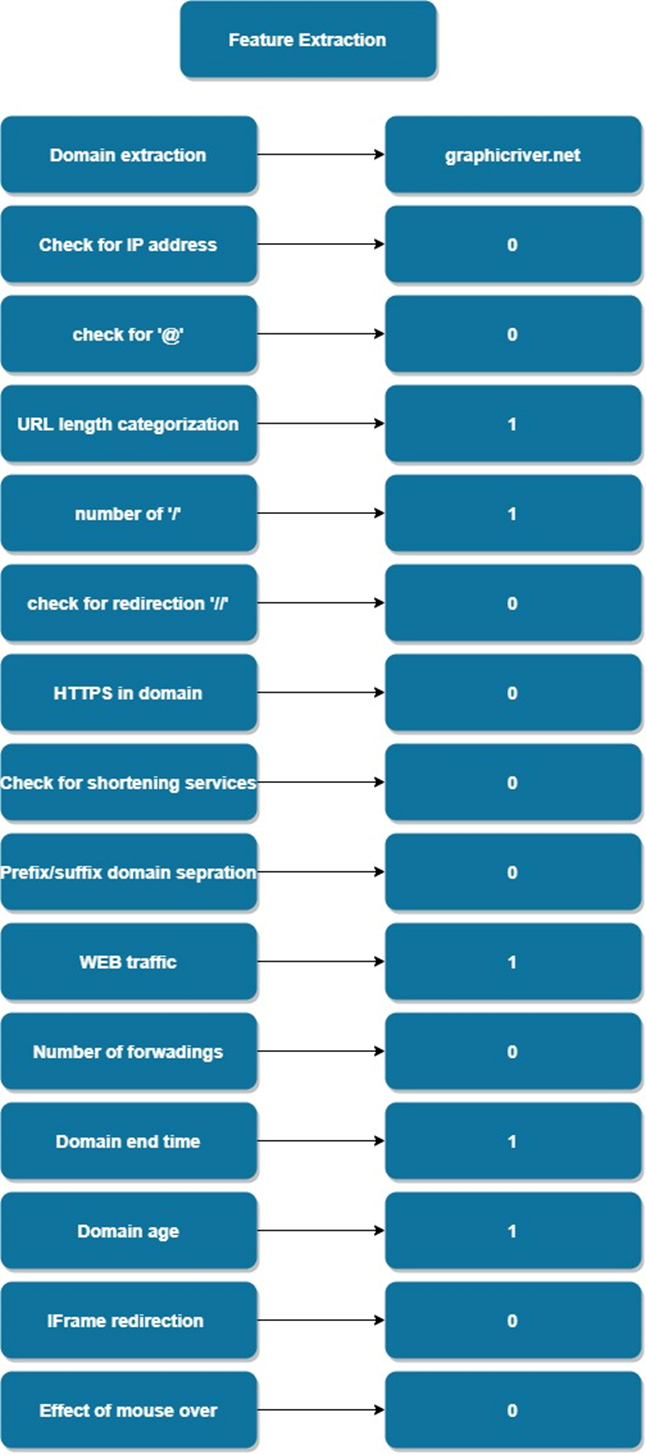
Domain Age (Domain Survival Time): This function offers data about the Domain Survival Time. It computes the discrepancy between creation and termination times. Long-running domains might be viewed as having a higher level of authenticity.

Domain End Time (Domain End): This function determines how much time has passed since the domain was terminated. A domain that is about to expire can be viewed as less trustworthy.

The IFrame Redirection (iFrame) function determines if the URL employs iFrame redirection. When an iFrame redirection is used, it may be a sign that the user is being taken to a separate website, which raises several red flags.

Status Bar Effect of Mouse Over (Mouse Over): This function examines the status bar effect of mouse over. The status bar's behaviour can reveal details about the legitimacy of a website.

Right Click Attribute Status (Right Click): This function determines the right click attribute's current state. The right click attribute's behaviour can reveal details about the legitimacy of a website.



Number of Forwardings (Web Forwards): This function determines how many forwardings are in the top of the form.

Creation of the Legal Dataset: Using the aforementioned functions, the characteristics were retrieved from the 5000 Legal URLs. The genuine dataset was updated with the returning values, which were either 1 or 0.

Phishing Dataset Creation: The phishing URL dataset was used to extract features using the same feature extraction techniques. The phishing dataset was updated with the returned results, which were either 1 or 0.

URL Data Creation: The urldata.csv file was made by concatenating the legitimate and phishing datasets. The model was then trained using this file.

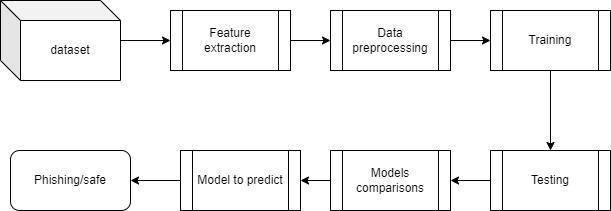
Important details about the structure and behaviour of the URLs were provided by the extracted characteristics, which were significant for identifying malicious URLs. The feature extraction features made it easier to comprehend the existence of IP addresses, URL length, URL depth, redirection, HTTPS token presence, and other significant characteristics.

Machine Learning Modelling: Our analysis was focused on this. It required transforming the uniform dataset into pertinent models. This sub-problem involved: (a) shortlisting machine learning models based on literature use for malicious website detection or similar tasks; (b)

reducing the list of shortlisted machine models to top models based on empirical performance in the literature related to malicious website detection and other well-known tasks; (c) choosing the metrics that would be used for training and validating the models; and (d) developing the machine learning models and training them on the training data, (e) saving the best result giving fine-tuned model real life predictions.

Machine Learning models used are:

1. Decision Tree Classifier: A tree-based supervised learning algorithm, a decision tree classifier generates a decision tree to make predictions. Due to its ability to manage intricate data linkages and produce understandable findings, it is important in the challenge of malicious URL identification. By creating a decision tree that predicts a URL's class based on its attributes, the technique can be used to identify whether a URL is harmful or not.

2. Random Forest Classifier: This ensemble learning approach builds numerous decision trees and averages their predictions to increase the model's accuracy and stability. Because it may improve accuracy and decrease the overfitting issue, it is important in the problem of malicious URL identification.

3. Multilayer Perceptrons (MLPs): Between the input and output layers of MLPs' feed-forward artificial neural networks are a number of hidden layers. Because of their ability to manage intricate non-linear correlations between the input attributes and the output class, they are important in the challenge of malicious URL identification. The intricate connections between the URL features and their class can be discovered using MLPs.

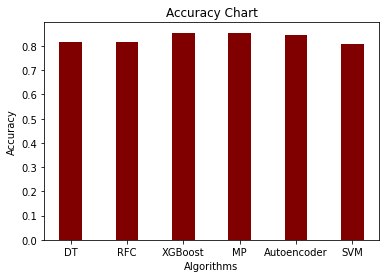
4. XGBoost Classifier: To increase prediction accuracy, XGBoost uses a gradient boosting tree technique to construct a tree-based model. Because it can handle big datasets, deliver results that are easy to understand, and increase accuracy by combining the predictions of other trees, it is important in the malicious URL identification problem.

5. Autoencoder Neural Network: Unsupervised deep learning methods known as autoencoder neural networks can learn to reassemble the input data. Due to their ability to extract crucial elements from URL data that can be used to create prediction models, they are critical in the challenge of malicious URL identification. To lower the dimensionality of the URL data and enhance the predictive model's performance, autoencoders can be utilised as a pre-processing step

6. Support Vector Machines are linear classifiers that locate the largest margin hyperplane to divide the data into many classes. Because they can handle high-dimensional data and produce reliable findings even in the presence of outliers, they are important in the challenge of malicious URL detection. By identifying the maximum margin hyperplane that divides the URL data into distinct groups, SVM can be utilised to develop a predictive model for the problem of malicious URL identification..

Figure 1. Flow diagram of our suggested system

1. RESULTS AND DECISION

The extracted features from the labelled data, which includes both harmful and helpful URLs, are then used to separate the data into training and testing sets [8]. The training data are supplied to different models including Multilayer perceptrons, XGBoost, AutoEncoder, Random Forest, Decision Tree, and SVM after being passed through various methods for feature extraction and labelling. Additionally, the trained model is evaluated using the training dataset, which contains features without labels, and the model's correctness is determined. The training dataset comes before the tested dataset in the process above, which is repeated for 80:20 data splits. The flow diagram for the experiment is shown in Figure 2. There are both benign and harmful URLs in the dataset. The three components of a URL are the protocol, domain name, and path. Numerous host-based features, lexical features, and site popularity features are derived from the URL. The following Source [8] is used to compile a collection of benign and malicious URLs.

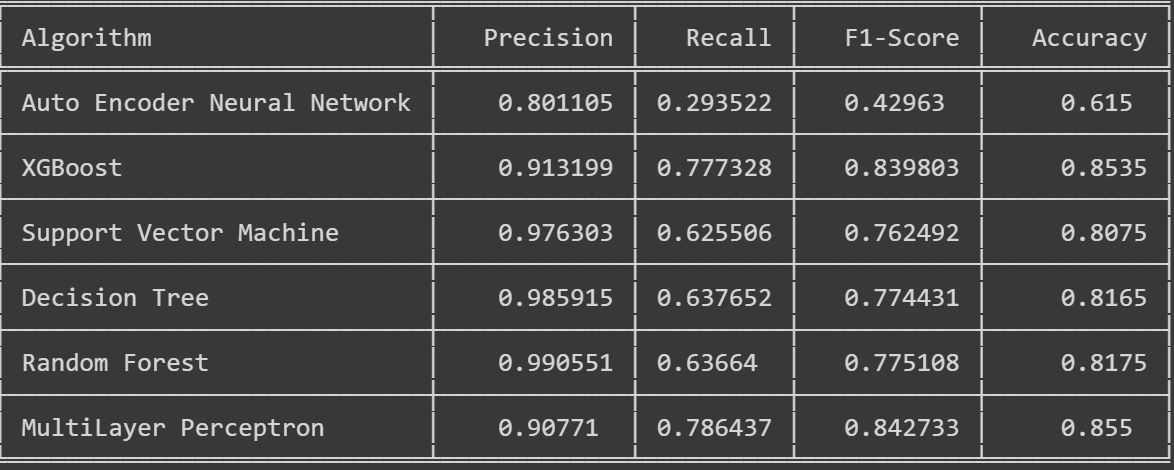
4.2 Model Analysis

The pandas library of Python is used to read the data, which are in CSV format. The training data's non-string columns are taken out. The data are delivered to SVM, XGBoost, Autoencoder Neural Network, Random Forest, Multilayer Perceptrons (MLPs), Decision Tree, and MLPs. Sklearn, a library for machine learning applications in Python, is available. The Decision Tree Classifier module is part of the Sklearn ensemble package. The input supplied to the Decision Tree function is the number of trees. The fit function, which accepts training data and output from feature extraction as inputs, is used to train the model. Utilising test data and the cross val score function cross validation module, the trained model's accuracy is evaluated. The result includes the test data and accurately anticipated values.

The dataset is split 80-20 into train and test sets for each classification model. of the data in a decision tree with a maximum depth of 5 were predicted accurately. Accuracy was 81.5% and 82.4% for Random Forest classification in training and testing data. Autoencoder Neural Network. accuracy was about 84.55% for Multilayer Perceptrons and XGBoost, and 80% for SVM.

1. F1 score is a statistic that combines recall and accuracy into a single value. It is referred to as the harmonic mean of recall and accuracy.

F1 Score: 2 + (precision - recall) / (precision + recall)

To offer a fair assessment of a classifier's performance, the F1 score considers both accuracy and recall. It is helpful when the dataset is unbalanced, meaning one class has much more data than the other.

1. Accuracy is a statistic that counts the number of occurrences that were successfully categorised.

Accuracy is equal to (TP+TN)/(TP+TN+FP+FN)

Accuracy offers a general assessment of a classifier's effectiveness. It calculates the percentage of correctly classified instances, where TP (True Positive) represents the number of positively predicted instances that actually occurred, TN (True Negative) represents the number of negatively predicted instances actually occurring, FP (False Positive) represents the number of positively predicted instances that actually occurred, and FN (False Negative) represents the number of negatively predicted instances actually occurring.

1. Precision is a statistic that counts the proportion of successfully anticipated positive cases among all positive instances that were expected.

Precision is equal to TP/(TP + FP).

Precision measures how well a classifier is able to anticipate positive cases. Instances of TP (True Positive), the number of properly predicted positive instances, and FP (False Positive), the number of mistakenly forecasted positive instances, are used to calculate the percentage of correctly predicted positive instances among all instances of expected positive instances.

1. Recall is a statistic that counts the proportion of positive cases that were properly anticipated out of all positive instances.

Recall is TP / (TP + FN).

Recall measures how well a classifier can anticipate positive cases. TP (True Positive) is the number of properly predicted positive cases, and FN (False Negative) is the number of mistakenly forecasted negative instances. It calculates the proportion of correctly predicted positive instances among all real positive instances.

1. CONCLUSION

The majority of online criminal operations are based on malicious Web sites. The risks posed by malicious websites are significant, and end users must not be allowed to access them. Users should refrain from clicking on these Uniform Resource Locators (URL). A number of machine learning algorithms, including Multilayer Perceptrons, XGBoost, AutoEncoder, Random Forests, SVMs, and Decision Tree are deployed on training dataset to solve the binary classification issue of detecting dangerous URLs. Additionally, it has been shown that the Multilayer Perceptrons and XGBoost classifiers outperform the others for the specific challenge.

1. REFERENCES
2. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Learning to detect malicious URLs. ACM Trans. Intell. Syst. Technol. (TIST) 2(3), 30 (2011)
3. Mansoori M, I Welch, Q Fu YALIH (2014) yet another low interaction honeyclient. In Proceedings of the Twelfth Australasian Information Security Conference 149: 7-15.
4. SahoO D, C Liu, SC Hoi (2017) Malicious URL Detection using Machine Learning: A Survey.
5. Ahmed Abdulghani Ali, Chua Xue Li (2018) “Analyzing data remnant remains on user devices to determine probative artifacts in cloud environment.” Journal of forensic sciences 63(1): 112-121.
6. A. Joshi, L. Lloyd, P. Paul Westin, and S. Seethapathy, "Using Lexical Features for Malicious URL Detection -- A Machine Learning Approach", 2019.
7. Detection of Malicious Social Bots Using Learning Automata With URL Features in Twitter Network, IEEE Transactions on Computational Social Systems Volume: 7, Issue: 4, August 2020)
8. Y. Zhou and G. Qiu, “Random forest for label ranking,” Expert Syst. Appl., vol. 112, pp. 99–109, Dec. 2018.
9. Detection of Malicious Websites Using Machine Learning Techniques, IJERTV11IS050004, Volume 11, Issue 05 (May 2022)
10. A. S. Manjeri, K. R., A. M.N.V., and P. C. Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," 2019 3rd International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 555-561, DOI: 10.1109/ICECA.2019.8821879.
11. An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset DOI: 10.1109/ACCESS.2021.3093094
12. Decisions Tree Learning Method Based on Three-Way Decisions DOI:10.1007/978-3-319-25783-9\_35
13. Firstfilter: A cost-sensitive approach to malicious URL detection in large-scale enterprise networks DOI:10.1147/JRD.2016.2558018
14. H. Kazemian and S. Ahmed, "Comparisons of Machine Learning Techniques for Detecting Malicious Web Pages", Expert Systems with Applications, vol. 42, no. 3, pp. 1166-1177, 2015. Available: 10.1016/j.eswa.2014.08.046.
15. M. Jordan and T. Mitchell, "Machine learning: Trends, Perspectives, and Prospects", Science, vol. 349, no. 6245, pp. 255-260, 2015.