iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42048

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Machine Learning System for Malicious WebsiteDetection using Concept Drift Detection

Chaitanya R. Vyawhare[1], Reshma Y. Totare[2], Prashant S. Sonawane[3], Purva B. Deshmukh[4]
[1, 2, 3, 4]*Department of Information Technology Engineering AISSMS Institute of Information Technology Pune, India*

Abstract: ***The rampant increase in the number of available cyber attack vectors and the frequency of cyber attacks necessitates the implementation of robust cybersecuritysystems. Malicious websites are a significant threat to cybersecurity. Miscreants and hackers use malicious websites for illegal activities such as disrupting the functioning of the systems by implanting malware, gaining unauthorized access to systems, or illegally collecting personal information. We propose and implement an approach for classifying malicious and benign websites given their Uniform Resource Locator(URL) as input. Using the URL provided by the user, we collect Lexical, Host-Based, and Content-Based features for the website. These features are fed into a supervised Machine Learning algorithm as input that classifies the URL as malicious or benign. The models are trained on a dataset consisting of multiple malicious and benign URLs. We have evaluated the accuracy of classification for Random forests, Gradient Boosted Decision Trees and Deep Neural Network classifiers. One loophole in the use of Machine learning for detection is the availability of the same training data to the attackers. This data is exploited by the miscreants to alter the features associated with the Malicious URLs, which will be classified as benign by the supervised learning algorithms. Further, owing to the dynamic nature of the malicious websites, we also propose a paradigm for detecting and countering these manually induced concept drifts. IndexTerms—URL Feature Extraction, Malicious Website Detection, Concept Drifts, Feature Vectors, Gradient Boosted Trees, Random Forest, Feedforward Neural Networks***

Keywords: (cyber attack, URL, Supervised machine learning, DNN, extraction)

I. INTRODUCTION

The internet is intended for gaining knowledge and staying connected, but there exist some negative elements which can disrupt this harmony. One such element is malicious programs. Malicious programs are pieces of code that are written with the intention of gaining access to a machine anddisrupting its performance. There are many ways by which malicious programs can infiltrate into a machine. [1] focuseson a common way in which malicious programs gain systemaccess i.e. by visiting malicious websites. Many a timesusers are not aware if the website they are visiting is genuine or bogus as they simply enter the URL of the website and in some cases, once they click enter, malicious code embedded enters the user's system and starts executing itself. Users are unaware that maliciousprograms have entered their system and started execution and as the systems performance starts deteriorating, they suspect presence of malicious programs but, by then the damage is already done. Identification of these URLs become necessary as they may harm systems or store passwords without user's information. The idea behind using Machine Learning techniques is to automate theprocess of classifying URLs. Human intervention is removed and overall performance is improved.

A variety of methodologies have been proposed and de ployed for detecting websites as well as mitigating the cyberattacks. One of the most widely used approaches is to maintain a centralized blacklist of suspicious as well as malicious sites, IP addresses, and domains [5]. The solution,though elegant and straightforward, is not scalable as it is tedious or impossible to validate and maintain a list of all these websites. To overcome the scalability issues, we propose a system for automatic feature extraction and real- time classifification using the input URL. we propose asystem for automatic feature extraction and real-timeclassication using the input URL for validating an unfamiliar or suspicious website. We are implementing an approach for classifying malicious and benign websites given their Uniform Resource Locator (URL) as input. Using the URL provided by the user, we collect Lexical, Host-Based, and Content-Based, URL-based, HTML-based, JS-based, IP-based and CA-based features for the website. These features are fed into a supervised Machine Learning algorithm as input that classifies the URL as malicious or benign. The models are trained on a dataset consisting of multiple malicious and benign URLs. With the classificationapproach Gradient Boosted Decision Trees.

Concept drift approach is implemented to increase the efficiency of model. If a concept drift is detected, we retrain our supervised learning model on the new data to improve the model's ability to detect URLs with manipulated features. In this paper, we define a novel approach for detecting concept drifts by computing the distance between old and new feature space. In this paper, we implemented a novel approach to detect any website given their URL, as malicious or benign and Retraining the model with concept drift approach.

©IJRASET: All Rights are Reserved

II. EXPLORATORY DATA ANALYSIS(EDA)

Exploratory Data Analysis (EDA) is an approach toanalysing data sets to summarize their main characteristics, often with visual methods. Following are the different steps involved in EDA.

Different steps are involved in EDA

1) Data collection
2) Data cleaning
3) Data pre-processing
4) Data visualisation

A. Phase 1: Data Collection

Data is the need of the hour, and its collection and analysis isthe base of any business and research success now.Datacollection is the process of gathering and measuringinformation from countless different sources. We used the following public blacklists to collect malicious URLs: PhishTank, a collaborative platform operated by OpenDNS to distribute and verify phishing websites and Malware Domain List (MDL). Benign URLs are retrieved from majestic.com .Our final dataset has a total of eighty thousandunique URLs, out of which forty thousand are malicious, and the remaining forty thousand are benign. Phishing URL's Website : "https://www.phishtank.com/". Benign URL's Website : "https://majestic.com/".

B. Phase 2 : Data Cleaning

Data cleaning is the process of ensuring that your data is correct and useable by identifying any errors in the data, or missing data by correcting or deleting them.Cleaning up datais the first and most important step, as it ensures the quality of the data is met to prepare data for visualization. While cleaning our dataset we have to deal with missing data or missing values, occur when no data value is stored/provided for the variable in an observation. To handle this missing values in datasets we can use mean , mode and median technique . If the dataset is small then you can removemissing values using manually.

C. Phase 3 : Data Pre-Processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. It includes normalisation and standardisation, transformation, feature extraction and selection, etc. The product of data preprocessing is the final training dataset. To generate understandable report we are using python libraries pandas which includes packages such as pandas_report and pandas_profiling. Which gives detail report of our datasets i.e number of missing values , distinct count , highly correctable features which will help to predict result as fast as possible.

D. Phase 3 : Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

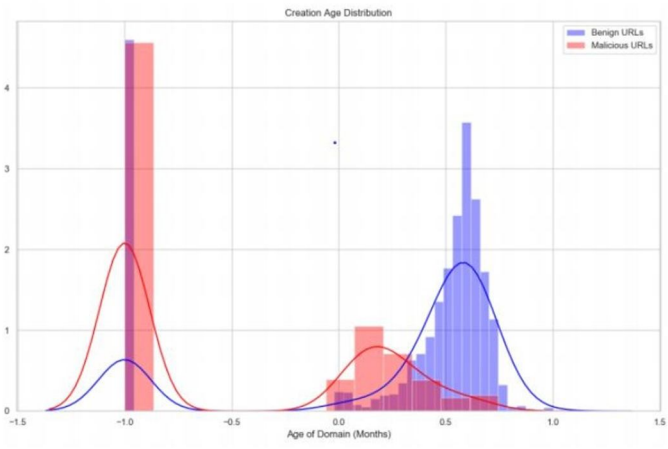


Fig. 1: Data Visualization(Create Age Distribution)

III. FEATURES EXTRACTION

In real life, all the data we collect are in large amounts. To understand this data, we need a process. Manually, it is not possible to process them. Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process them. So Feature extraction helps to get the best feature from those big data sets by select and combine variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with the accuracy and originality. We categorize the URL features into three groups: lexical, host-based, content-based and domain based features. whereas lexical based features and domain based features which have been known to improve the performance of the classifier.

A. Lexical Features

The main motivation of utilizing URL-based featurescomes from the fact that a malicious URL can have a resembling name when compared to a benign URL. Since a textual name cannot be directly utilized in models, lexical features focus on extracting the statistical properties such as length, of sub-domains and many more from a URL.



Below mentioned are all lexical feature which are extracted from the URL (Uniform Resource Locator) string. most of the feature plays crucial role in identifyingthe malicious URL.

Features Name	Description
URL Length	Length of the Uniform Resource Locator
Host Length	Length of host name
Host Token Count	Number of tokens in the hostname (delimited by '.')
Path Length	Length of the path
Number of Symbols	Number of special characters like
IP Present	Check IP is present or not
Query Length	Length of query
Is sub domain available	Check for sub domain

Table 2: List of Lexical based features utilized in this work

B. Host Based Features

In addition to URL-based, HTML-based and JavaScript features, we extract Host based features whichare dependent on the domain of the URL. If the URL does not contain any domain or we are unable to extract afeature, then we set the default value as -1. In this work, we extract the Host-based features using the WHOIS information of the domain. Specifically, we utilize geographical, creation, expiration, updation date and many more as features for the classification purpose.

GeoIP python library which provides access to GeoIP databases. whereas from python-geoip we can extract so many features like geographic location, locale code, continent code, continent name, time-zone etc.

Features Name	Description	Type
Host age create mon	Counts number of months since domain was created	Integer
Host age exp month	Counts number of months since domain was expired	Integer
Zip code	Provides 1 or 0 whether zipcode of domain is available	Integer
City	Provides 1 or 0 based on whether city for domain is available	Binary
Status	Returns status of domain	Binary
Country	Provides 1 or 0 based on country for domain is available	Binary
Whois server	Provides 1 or 0 based on whether whoisserver information for domain is available	Binary
Location	Provide location of host	Integer
Autonomous System Number	Provide ANS number	Binary

Table 1: List of Host based features utilized in this work

C. Content Based Features

1) *HTML Features:* Described HTML features as ”heavy weight” because these features depend on the actual web- page content. It can be seen from Figure 3.1 and 3.2, a lot ofinformation can be extracted from the website which can models' accuracy in distinguishing malicious URLs from benign ones. Since most of the website content is textual, lexical and statistical properties of a page such as length of document, average length of any word, number of each of the special characters(.,-, ,=,,@) etc. are utilized. For extraction of HTML based features from HTML documentation source code we are using Beautifulsoup library . It is a python library that is used to extract data fromHTML and XML files.

In addition to above mentioned HTML features, proposed below features which can help to distinguish between malicious and benign one.

Features Name	Description	Type
Html numbr iframe	Counts number of months since domain was created	Integer
Html numbr small elements	Counts number of months since domain was expired	Integer
html script len	Counts number of script tags	Integer
Html nmbr whitespaces	Counts number of white-spaces inside body of HTML	Integer
html nmbr ext domain	Counts number of elements whose source is on external domain	Integer
html link in link	Counts number of links in the HTML document	Integer
html document len	Counts number of total words in the document	Integer
html content nmbr hyphens	Counts number of hyphens (-) in the body of HTML	Integer
html srvr f ile handler	Provides Server File Handler	Integer

Table 3: List of Host based features utilized in this work

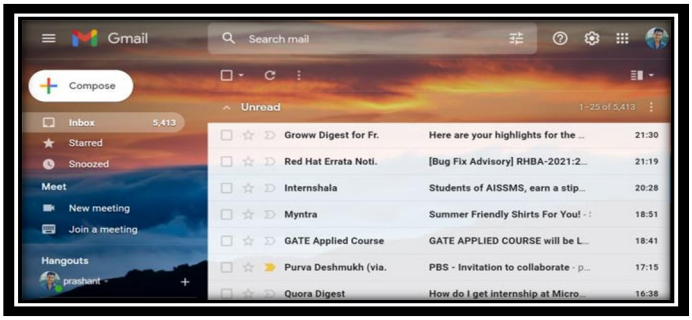


Fig 2 : Webpage

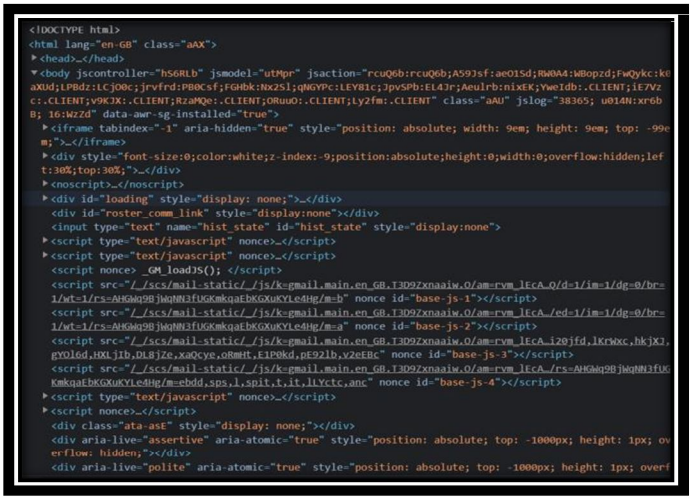


Fig 3 : Webpage HTML source code

2) *JavaScript Features:* JavaScript (JS) features can be extracted either from JS files(.js) or from actual script embedded inside script tag in the HTML code of any website. Researchers claim that JS code is a rich source of malware activity where hackers try to exploit functions suchas eval(), escape(), unescape() etc. . Similar to HTML features, JS features can be categorized as: lexical and statistical. For extraction of javaScript features we are using Esprima library. It is a high performance, standard-compliant ECMAScript parser written in ECMAScript (also popularly known as JavaScript). It is used to perform lexical analysis (tokenization) or syntactic analysis (parsing) of a JavaScript program.

Features Name	Description	Type
js count eval	Counts number of eval() function	Integer
js count setTimeout	Counts number of setTimeout() functions	Integer
js count unescape	Counts number of unescape() functions	Integer
js count escape	CCounts number of escape() functions	Integer
js count concat	Counts number of concat() functions	Integer
js count substring	Counts number of substring() functions	Integer
js count documentadd-dEventListener	Counts number of documentadd-dEventListener() functions	Integer
js document len	Counts number of characters in the script	Integer
js nmbr at	P Counts number of at(@) in the script	Integer

Table 3: JavaScript features utilized in this work

From Table 3, it can be inferred that all the features are numerical type, however, while extracting the data if the ping request fails or we are unable to fetch any feature then default value is -1

IV. PROPOSED FRAMEWORK

In this section, we defifine the design of our proposed framework which classififies the website as malicious orbenign. Further, the system is equipped with the ability toretrain the supervised machine learning model in thepresence of concept drifts. Our system consists of threemodules, namely feature extraction, supervised learning andconcept drift detection. Basically to train the machinelearning model, Data analysing is needed and thenperforming operations on that.

The incoming URL is an input to our system.As we get an input URL, firstly the feature extraction process takes place,that is, host-based,lexical,CA, domain-based,IP address,and many more features are extracted.

The main part of our system is the supervised learning model, which detects whether the input URL is benign or malicious. For this, before hand, we have trained our model, with Machine learning algorithm and Data mining. As the features are extracted of the input URL, these features are fed as input to supervised learning model, After that, the model clasififies the input URL as benign or malicious.

Before that, these features are transmitted to training data setof the model, And then to Concept drift detection model, if aconcept drift is detected, then it is removed by using the Heterogeneous Euclidean Overlap Metric (HEOM) formula method which increases the accuracy of the output and acts an quality measure of the system. And thus, the output, URLas malicious or benign is detected.

A. Feature Extraction

This module makes use of opensource python libraries such as Requests, WHOIS, BeautifulSoup and open-source GeoIP database to extract features from a given URL. All the features given in Table I are obtained using thismodule.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

B. Supervised Learning Model

This module is responsible for training the supervised learning model followed by predicting the degree of maliciousness for unseen URLs. We analysed threealgorithms, namely eXtreme Gradient BoostedTrees, Support vector machine and k-nearest neighbors as candidates for our learning model.

1) *eXtreme Gradient Boosted Trees:* Gradientboosting is a machine learning techniquewhich uses an ensemble of decision trees to form the prediction model. It sequentially builds the model such that the succeeding predictors can learn from the mistakes of the preceding predictors
2) *Support Vector Machine:* support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.
3) *K-nearest Neighbors:* k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical unitsor come in vastly different scales then normalizing the training data can improve its accuracy dramatically.

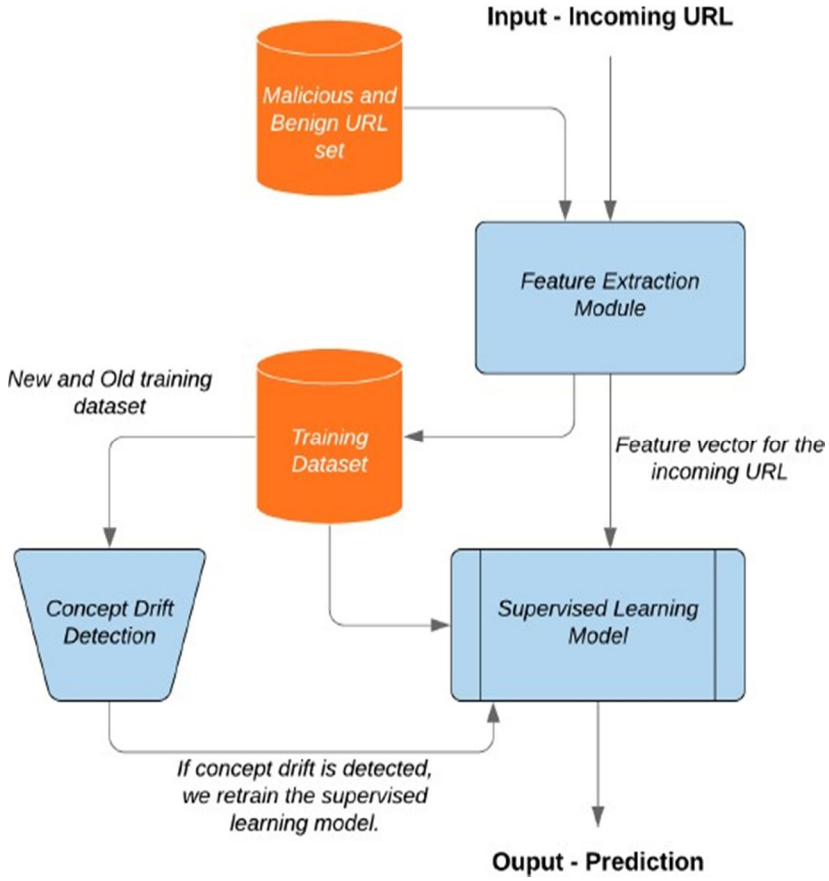


Fig 4: System Architecture

C. Concept Drift Detection

We propose an algorithm to detect concept drifts in real time efficiently. During concept drift detection, we againcollect the feature vectors for all the malicious and benign URLs in our training data set. This algorithm finds the difference in data distribution between the old training data and the newly collected feature vectors dataof input URL. For every malicious URL and benign URLin the new training dataset, we fiind the least distant malicious URL feature vector and the least distant benignURL feature vector in the old dataset. The Heterogeneous Euclidean-Overlap Metric (HEOM)is used for calculating the distance between two feature vectors.If the average ofall the distances calculated for this time period is more than a pre-set threshold value, then the drift has occurred.Once concept drift is detected, we retrain our supervised learning model on the new data to improve the model's ability to detect URLs with manipulated features.

©IJRASET: All Rights are Reserved

D. Concept Drift Detection Algorithm Flow Chart

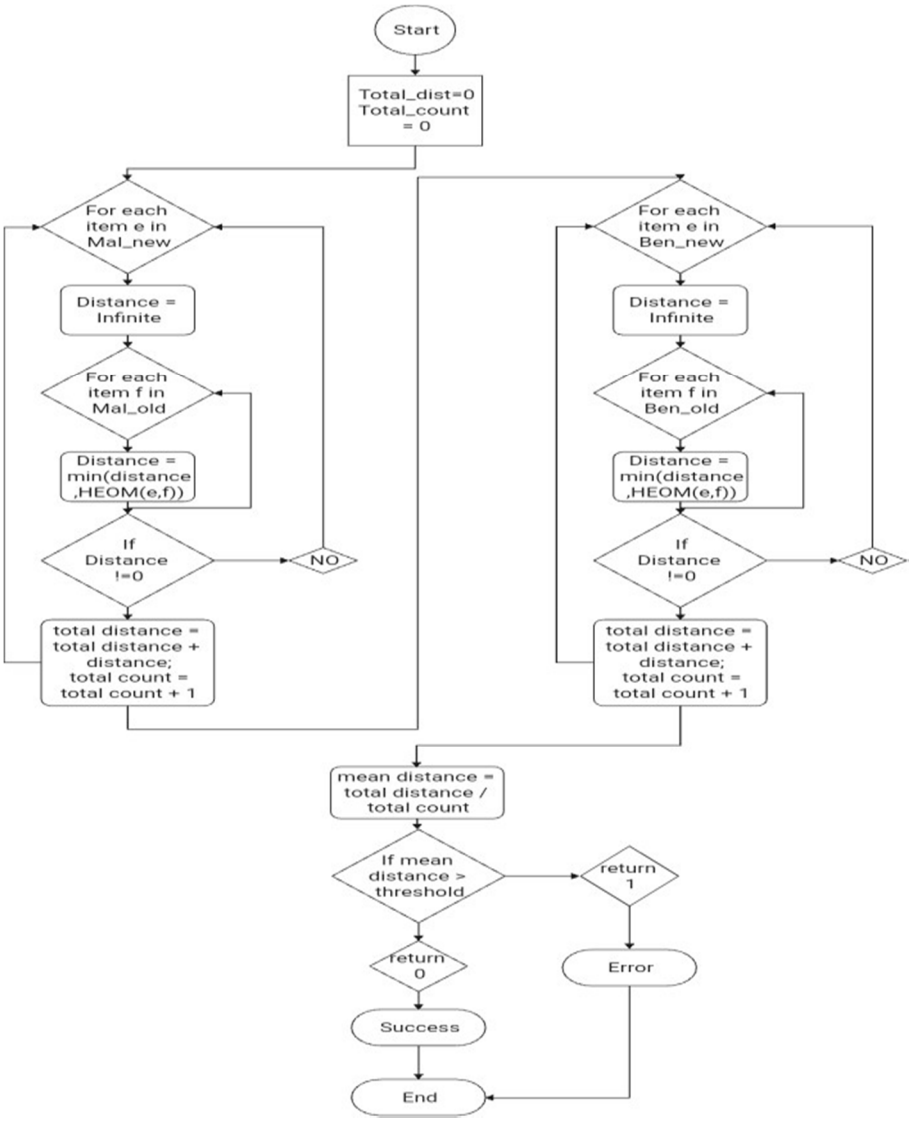


Fig 5: Concept Drift Detection Algorithm Flow Chart

E. Heterogeneous Euclidean-OverlapMetric(HEOM)

This Formula technique is employed for calculating the smallest amount of distance between 2 feature vectors e.g. malicious old vector and malicious new vector as shown in flow chart. If the average of all the smallest amount of distances are calculated for now and is greater than quite a pre-set threshold, then the concept drift has occurred. The HEOM appearance at every attribute of the instance (x, and y) and depending on the sort of knowledge with following computations: If categorical — returns zero if the attribute is of an equivalent category, one otherwise If numerical — computes the space using the normalized Euclidean metricIf missing — returns 1 Then, we can get a result vector that stores the distance for each attribute. To obtain the final result, the following equation can be used:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a,y_a)^2}$$

HEOM has been known to perform well with vectors consisting of both numerical and categorical attributes, hence it fifits perfectly with the dataset discussed in the paper. Once concept drift is detected, we retrain our supervised learning model on the new data to improve the model's ability to detect URLs with manipulated features. Occurrence of a concept drift can also be validated by a steep fall in the accuracy of our classification model on our validation set

V. PERFORMANCE ANALYSIS

The XGBoost algorithm is used to verify the accuracy and stability of malicious URL classification models. In the experiment herein, the original 41 features and the selected Top 17 features were analyzed separately, which is the most efficient number of features for reducing the complexity of training model is found in the features filter process.

Following the training of the XGBoost classification model, the dataset includes about 13,027 URLs of benign websites and 13,027 URLs of malicious websites. Ten-fold cross-validation is used to train the malicious URL classification model with XGBoost. Finally, XGBoost and ANOVA are used to reduce the number of dimensions of features, and determine the best number of features of training data to optimize the model.

Cross-validation is a method of evaluating a predictive model by dividing the original sample into a training set and a test set of the model. This study applies 10-fold cross- validation, that main dividing training set into 10 parts.

Taking rotation of 1 different part as a test set and the remaining 9 parts as a training set as shown in Figure 6. This study individually entered into four classic machine learning algorithms (KNN, Decision Tree, SVM, XGBoost) and the performance of different algorithms is compared, the trained and 10-fold cross-validation comparison table is shown in Table 5. Using accuracy as the main standard, it can be found that the Tree-based algorithms perform better than the others, and XGBoost is better suited for this task than the Decision Tree algorithm. Therefore, using the top 17 important features on the XGBoost algorithm, experiments were performed in a plus-one in-loop manner, and the results are shown in Figure 7. This experiment was conducted to reduce the complexity of the model and maintain a higher accuracy. From the figure, it can be seen that the accuracy of XGBoost reached 99.98% when the ninth feature was added and started to decrease when the tenth feature was added. Therefore, it can be concluded that this data set on the XGBoost classification model can achieve 99.98% accuracy using only the first nine features, with high classification performance and efficiency.

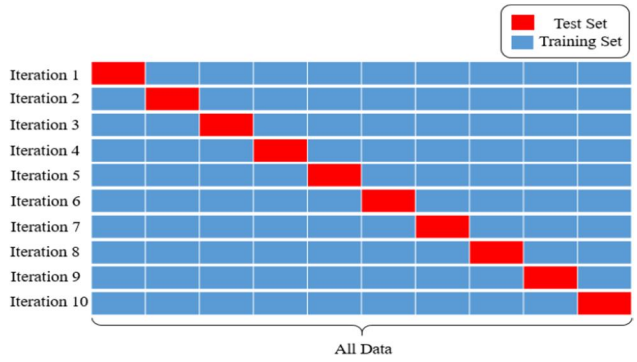
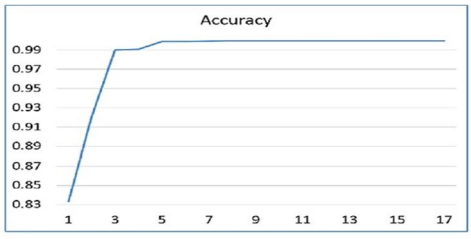


Fig 6. 10-fold Cross-Validation

Table 1: COMPARISION OF MACHINE LEARNING ALGORITHM

Algorithm	Accuracy	Precision	Recall	F1_Score
KNN	99.25%	99.50%	99.01%	99.26%
SVM	98.74%	100%	97.50%	98.73%
XGBoost	99.99%	100%	99.99%	99.99%



Features	1	2	3	4	5	6	7	8	
Accuracy	0.833	0.921	0.99	0.991	0.9987	0.9989	0.9994	0.9995	
Features	9	10	11	12	13	14	15	16	17
Accuracy	0.9998	0.9996	0.9996	0.9997	0.9997	0.99976	0.99973	0.99973	0.9998

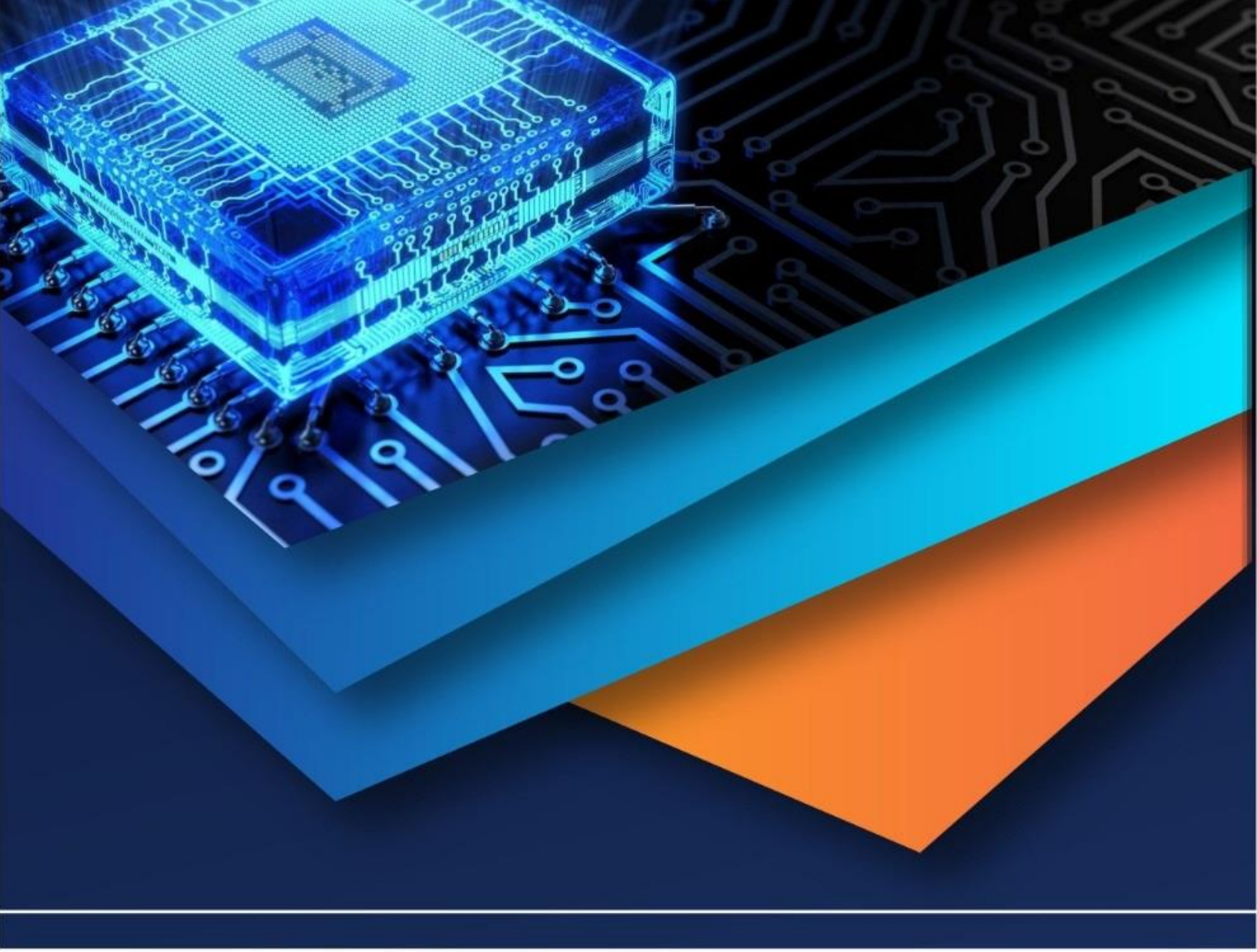
Fig 7. Accuracy of XGBoost with Top 17 features

VI. CONCLUSION

In this paper, we have proposed a robust and novel approachfor assisting users in detecting malicious and harmful websites and URLs. The dataset used for training and testing has been curated from a variety of sources. Our algorithm achieves a superior accuracy of 96.4dataset. We have proposed a novel approach for detecting and mitigating attempts by attackers for circumventing malicious website detection algorithms. Results show that the proposedalgorithm is able to successfully detect concept drifts.

REFERENCES

[1] S. Singhal, U. Chawla and R. Shorey, "Machine Learning Concept Drift based Approach for Malicious Website Detection," 2020 12th International Conference on Communication Systems Networks, pp. 582–585, 2020
[2] Yu-Chen Chen, Yi-Wei Ma, Jiann-Liang Chen "Intelligent Malicious URL Detection with Feature Analysis" ©2020 IEEE
[3] A. K. Singh and N. Goyal, "A comparison of machine learning attributes for detecting malicious websites," 2019 11th International Conference on Communication Systems Networks (COMSNETS)
[4] A. K. Singh and N. Goyal, "A comparison of machine learning attributes for detecting malicious websites," 2019 11th International Conference on Communication Systems Networks (COMSNETS)
[5] T. Guolin, Z. Peng, L. Qingyun, L. Xinran, Z. Chunge, and D. Fenghu, "Adaptive malicious url detection: Learning in the presence of concept drifts," 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications.
[6] G. Chakraborty and T. T. Lin, "A url address aware classification of malicious websites for online security during web-surfing," 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).
[7] Tan G, Zhang P, Liu Q, Liu X, Zhu C et al. "Adaptive malicious URL detection: Learning in the presence of concept drifts," 2018 17th IEEE International Conference on Trust, pp. 737–743, 2018.
[8] AS. Manjeri, K. R, A. MNV and PC, Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," Third International Conference on Electronics Communication and Aerospace Technology, pp. 555–561, 2019.
[9] F. Alkhudhayr, S. Alfarraj, B. Aljameeli and S. Elkhdiri, "InformationSecurity: A Review of Information Security Issues and Techniques," 2nd International Conference on Computer Applications Information Security, pp. 1–6, 2019.
[10] A. Bhagwat, S. Dalvi, K. Lodhi and U. Kulkarni, "An Implementionof a Mechanism for Malicious URLs Detection," 2019 6th International Conference on Computing for Sustainable Global Development, pp. 1008–1013, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)