iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** V **Month of publication:** May 2020

DOI: http://doi.org/10.22214/ijraset.2020.5447

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: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

Detecting Malicious URL using Machine Learning: A Survey

Sunitha M Nair[1], Divya Prasad K H[2]
[1]*M. Tech. Student,* [2]*Assistant Professor, Dept. of Information Technology, Government Engineering College, Barton Hill, Kerala, India*

Abstract: ***Malicious Uniform Resource Locator/malicious websites are a high threat to cyber security. Malicious URLs host uninvited content like malware, spam, drive by downloads phishing, etc. Users become victims of scams like financial loss, thieving of personal data, malware installation, and causes losses of millions of dollars every year. There is a need to detect those threats in a very efficient and timely manner. Several studies have examined different techniques to handle the problem; the foremost used approach remains blacklisting. The most obstacle to using blacklist is that the difficulties in maintaining an up-to-date list of URLs. So here we proposed the Machine learning approach to detect the malicious URLs. We also discussed various methods for malicious URL detection, feature representations, and finally discussed various algorithms for the classification and feature extraction.***

Keywords: Malicious URL Detection, Machine Learning, Blacklisting, Spam, Cyber Security, Malware.

I. INTRODUCTION

The Internet has been growing at new rates. There are a variety of attacks on the web. Most of the time, through the web malicious software also referred to as malware, or attacks is propagated. These malicious softwares delivers malicious content on the web. Currently, there are a number of approaches to the detection of dangerous websites on the web. One of the most common methods used by several antivirus systems and online services is the blacklisting approach. But, due to some limitations, the blacklisting approach is not sufficient to detect various attacks that are not blacklisted. The users surf the internet by typing a URL or an address. This may led to a variety of attacks like drive-by-download [1], which is the transfer of malicious software system (malware) onto your laptop or computer or mobile device. Watering hole, that may be a computer attack strategy, within which the victim is part of a group or a cluster. During this attack, the attacker usually observes the website the victim uses and they may launch attacks on it. The URLs used to implement the attacks are supposed malicious URLs. URL is the abbreviation of Uniform Resource Locator, also termed an internet address. A URL has two main components.

Protocol Identifier - It defines the protocol that can be used to search the resource. For example, consider the URL or Uniform Resource Locator for Google, “http://www.google.com". Here "http" is referred to as the Protocol identifier.

Resource Name – It is the complete address of the resource. For example, the address or Uniform Resource Locator for Google, "http://www.google.com". Here "google.com" is the Resource name. The protocol symbol and the resource name are separated by a colon (:) and two forward slashes (//). The associate example is shown in Figure 1.

To detect malicious URLs several antivirus teams proposed blacklist technique. Blacklist approach [2] includes a database which consists of a number of URLs that are already malicious. This information is compiled in a timely manner. This technique is fast due to a simple query since only a database lookup needs to be performed. We know millions of new URLs are generated every day so it is very difficult to keep an updated list of URLs.

The next method is Heuristic approach [4]-[6]. These are same as that of blacklist. Here instead of using a database of URLs, attacks are identified. To each of the recognized attacks, a signature is allotted. The issue with this approach is that only some common attacks are considered and signature is allotted. This technique will not consider all the attack possibilities. The next technique to notice malicious URLs is Machine Learning [7], [8]. This technique can give better results as compared to blacklist and heuristic approaches.

This method analyses the URL i.e. each and every character, numerals, Special characters, etc....From that it will extract some features regarding the URLs. That features can be further used to train a model. So whenever a new URL is encountered it is fed to the model and it will classify by extracting the features. So here we proposed different machine learning techniques and algorithms for Malicious URL Detection. It extracts the features of the URLs, learns a prediction model to classify a URL as malicious or benign.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

II. METHODS FOR DETECTING MALICIOUS URL

The malicious URL detection [3], [9-13] is an emerging issue. Now several services have used machine learning to resolve this problem. Here we discusses various methods that can be used to detect malicious URLs. They are (i) Blacklisting Approach (ii) Heuristics Approach and (iii) Machine Learning Approach.

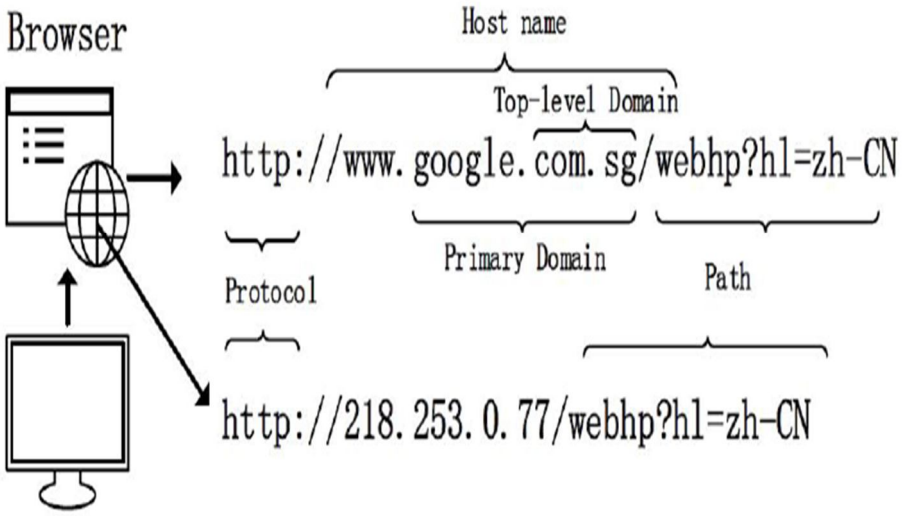


Fig. 1 Illustration of a URL

A. Blacklisting Approach

Blacklisting approaches [2] are the most commonly used technique for the detection of malicious URLs. Here it maintains a database which consists of a large number of URLs that are malicious. This approach will check the database periodically when a user enters a URL. If a match is performed then a warning message is generated in order to notify that the URL that is entered by user is malicious and it may launch attacks in the systems. If no match exists then it is considered as benign one. The database is compiled regularly whenever a new malicious URL is encountered. Blacklisting is very fast since we just need to check a database. The issue with this approach is that we know a millions of URLs are generated every day. So it is very difficult to keep a list of the entire URLs. Thus this approach will fail in noticing new attacks. Several attackers may use a number of techniques that can fool the legitimate users. They usually modifies [14]-[16] the URL by using larger hostnames instead of short names or ip addresses, also they modifies the URL by using wrong spellings that seems same as original ones. Using these methodologies attackers hide the maliciousness of the URLs. Another limitation is that it is not suitable for new URLs. This approach only searches for the URL in the database. When a new URL is encountered making it impossible for them to notice new threats.

B. Heuristic Approach

Heuristic approaches [4]-[6] are an improvement to the blacklisting approach. Here a blacklist of signatures is formed. Some of the common attacks are identified. For each of them a signature is allotted. In this approach when a user visits or types a URL then a signature is allotted to it. Then it is analysed with the available set of signatures. If any match exists between the two then that URL is said to be malicious otherwise not. One of the disadvantage of this approach is that it is designed for a limited number of attacks i.e. only some most occurring attacks are identified and the signature [17] is allotted. So it is not possible to consider all the attack types. Also this technique cannot predict in case of new URLs.

C. Machine Learning Approach

Machine learning approaches [7], [8] analyses different information regarding the URLs and its WebPages. It analyses several informations from the name, ip address, domain name, host name etc.. These informations together are known as features. These features are used to train a model. The model will predict whether a URL is malicious or not. We train the model for both malicious and benign URLs. Whenever a user visits a new URL then the features regarding the URL are collected and then it is fed to the classification model and it will predict whether the URL is malicious or not.

©IJRASET: All Rights are Reserved

There are two forms of features that may be used: static features and dynamic features. In static analysis, the URLs are analysed without executing the contents. Features are extracted that include the host information, contents of HTML and JavaScript. The features that are extracted for malicious and benign ones are different. This help in the classification to obtain a better result. Most of the machine learning technique use static analysis. Dynamic analysis techniques monitor the behaviour of the systems that are potential victims. Dynamic analysis techniques have inherent risks and are tough to implement and generalize. A framework for malicious URL detection is shown in figure 2.

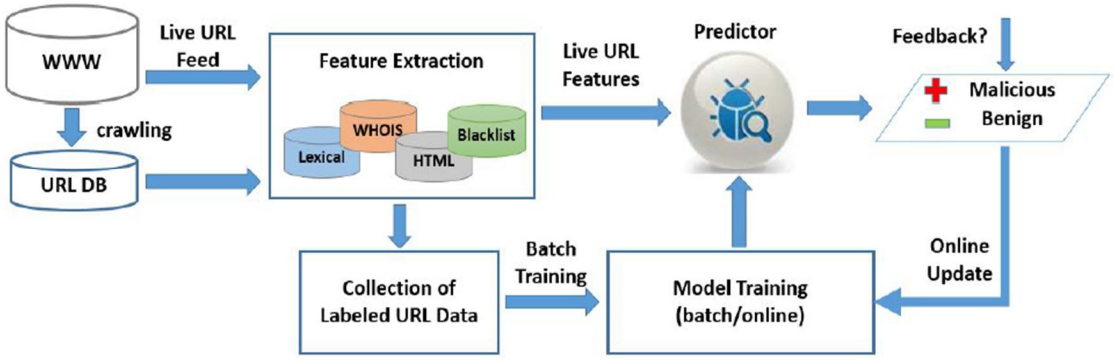


Fig. 2 A framework for Malicious URL Detection

III. FEATURE REPRESENTATION

To detect malicious URLs, a variety of features have proposed that can be used to provide useful information. The features include:

A. *Lexical Features*

Lexical features are features obtained from the URL name itself. i.e., Based on how the URL looks it should be possible to identify whether a URL is malicious or not. Attackers made modifications in the URL name to make it look like benign URLs. So lexical feature only cannot determine the maliciousness of a URL. It can be used along with some other features of the URL such as its host name features or content features and so on. It can improve the performance of the model. Also the URL name is not used directly as they appear in machine learning instead they are processed and then the features are extracted and then it is fed to the model. Lexical features can be divided into two categories namely Traditional lexical features and Advanced lexical features.

Traditional Lexical Features: Traditional lexical features include the common properties of the URL i.e. obtained from the URL name. This include the length of URL, number of dots in it, number of special characters, length of its domain name, protocol used, TLD used etc...The URL consists of a number of strings such that they are separated by using some special characters which can be '=', '/', '_', '-', '?' etc...The strings along with these special characters comprise a word. By using these words, a dictionary was constructed. Each word in that dictionary is considered as a feature. During the feature extraction , if any of the words in the dictionary is present in the URL, then the value of feature would be 1 and if not present the value would be 0.This method is known as the bag of words model.

Advanced Lexical Features: Researchers have proposed several advanced lexical features to collect more informative features about a URL. The motivation of collecting new or advanced lexical features is that to become free from obfuscation [18] of the attackers.

B. *Host Based Features*

Host-based features can describe "where" websites are hosted i.e. the country, location, time of hosting etc.., "who" own them i.e. the person or organization that created it and "how" they're managed i.e. who can access these resources. These are some of the properties of the hosts [19], [20] that are identified by the hostname part of the URL.

1) *IP Address Properties:* This explains the features of the IP address of the URL. IP addresses are a set of 0s and 1s. It is made of 32 bit. Each four sets comprising 8 bits. IP address properties indicate whether the IP address [37] is used in the URL, whether it is present in any black list etc..
2) *WHOIS Properties:* The word WHOIS indicates "Who is responsible for the Domain name?" It indicates who created the domain name, which country it was created, time of creation etc... This can be used to get the information regarding the domain name. WHOIS is a database that stores the information of registered users of an internet resource such as Domain name, IP etc. It is a query and response protocol. It stores all the contents in readable format. When the user requests it is delivered.

©IJRASET: All Rights are Reserved

3) *Domain Name Properties:* Domain names are used to identify IP addresses. For example, the domain name google.com contains a dozen IP addresses. Domain names are used in URL to identify a particular webpage. Users can use either IP address or the domain name to access web pages or resources.
4) *Geographic Properties:* This indicates the location of the IP address i.e. in which continent/country/ city does the IP address belong? What is the speed of the connection?

C. Content-Based Features

Content-based features are obtained by downloading the webpage of the URL. Content-based features are very large since the entire web page need to be extracted. The prediction model would be good and accurate since it provides more information about a URL. In case of the URL based features or lexical features fail to detect the maliciousness [21] of the URL, then this content based features can be used since it gives a lot of information as compared to URL-based and host-based features The content-based features can be categorized into three segments: HTML Features, JavaScript features, Visual Features.

1) *HTML Features:* Researchers extracted [22], [23] the webpage of several URLs in order to determine whether a URL is malicious or not. They extracted the HTML web pages of the URL. They analysed it and features are extracted from it. This features can be further used for training the prediction model. Some of the features include length of the entire webpage or document, the number of Null characters, presence of string concatenation [36] ,length of each word in the document, number of lines in the document, presence of link to other pages or not etc....They arrived into a conclusion that the malicious codes hide behind the string concatenation or large words in document.
2) *JavaScript Features:* In order to hide the malicious codes a number of JavaScript functions are used by attackers. The clients are not aware of this installation of malicious codes. Some of the examples of JavaScript functions they used are escape (), unescape (), exec (), and search () functions. Sometimes including long strings, large number of string assignments, number of string tags also hides some malicious contents in the URLs.
3) *Visual Features:* Visual features use the visual similarity of the web pages. They compare the web pages with that of the protected pages that are already present. This will help to identify whether that web page host any malicious content or not. Some examples are CCH [25], OCR [26].

IV. ALGORITHMS USED FOR MALICIOUS URL DETECTION

Several machine learning algorithms can be used to solve malicious URL detection. The URLs are analysed and feature vector is constructed. Then it is fed to the model for training. The learning algorithm can be classified into two namely Batch Learning Algorithm and Online Learning Algorithm.

A. Batch Learning Algorithm

Consider a set of T URLs i.e. URL from the training data. Let Yt = (1, -1). Yt represents the class label which can be malicious or not, spam or not, phishing or not etc....Here the class label y = 1 indicates a malicious URL and y = -1 indicates a benign URL. The URLs are analysed and it is mapped to feature vector. Then any of the learning algorithms can be used to train a prediction model. We train both malicious URLs as well as benign URLs. An illustration is given in figure 3. The most common batch learning algorithms that have been applied for Malicious URL Detection are: Support Vector Machines, Logistic Regression, and Decision Tree.

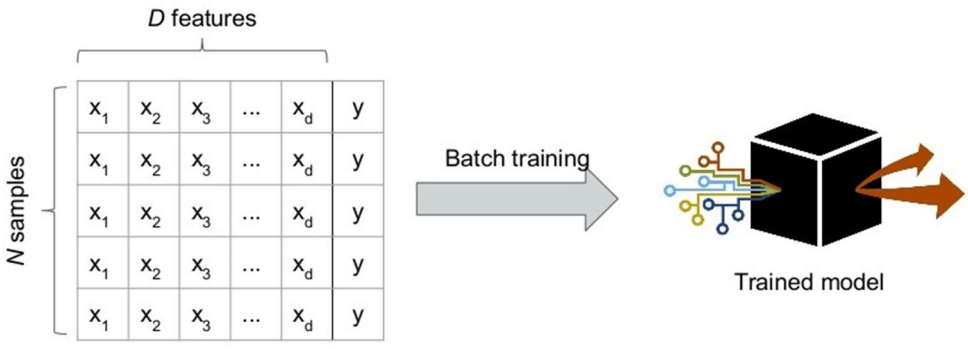


Fig. 3 Illustration of Batch Learning

©IJRASET: All Rights are Reserved

1) *Support Vector Machines:* Support Vector Machine (SVM) [27] is one of the most popular supervised machine learning [30] algorithms. It is mostly used in classification problems. Here, we have a number of data items i.e. here, a set of URLs. We plot each of the URLs data items in an n-dimensional space; n represents the number of features. Here the goal is to find a hyperplane [29] that can classify the data items or the URL accurately. In our case we want to classify the malicious URLs in one coordinate and the benign ones in other coordinate. In case of labelled data this approach is best and in case of unlabelled data we have to use unsupervised learning [28]. In case of unsupervised learning, the similar data's are grouped to form a cluster. Whenever a new data item comes then it is checked against different clusters. It is added to the cluster having greater similarity. The support vector clustering algorithm can be used to categorize unlabeled data. The main goal of the SVM is to train a model that classifies a newly generated data item into correct coordinate. It creates a partition of features into two categories. Based on the features or similarities of the object it places an on object just above or below the hyperplane. Let us consider some scenarios:

a) *Scenario-1:* Consider we have three hyperplane namely hyperplane A, hyperplane B, hyperplane C. The data items are star and circle. Now the goal is to identify the correct hyperplane that divides the data items. From fig 4, we can notice hyper-plane B correctly divides [24] the two classes.

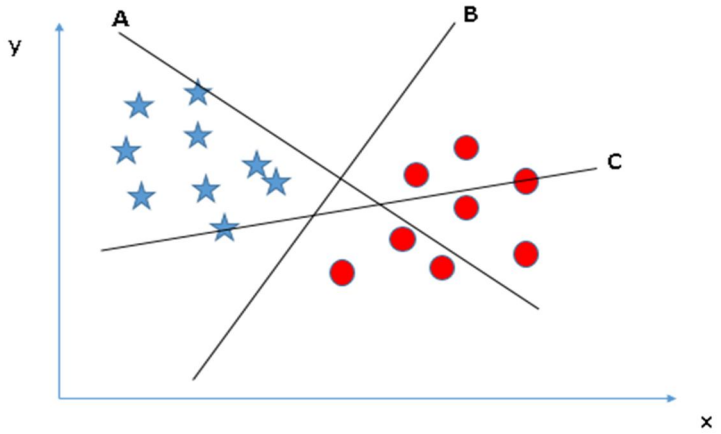


Fig. 4 Support vector machine hyperplane scenario - 1

b) *Scenario-2:* Consider three hyper-planes A, B and C. In figure 5 we can notice the three hyperplane equally segregates all the data items. So our goal is to find which is the best hyperplane that divides all these data items. For that find the distance between the hyperplane and the data item just near to it. The distance can be termed as the Margin. Always choose the hyper plane with higher margin. The reason of choosing the hyper plane with higher margin is that if we choose the lower margin it will led to miss classification of the data points. So in figure 6 we can notice the hyperplane C has higher margin as that of hyper plane A and hyperplane B. So C is considered as the right hyperplane. Scenario 2 is illustrated in figure 5 and figure 6.

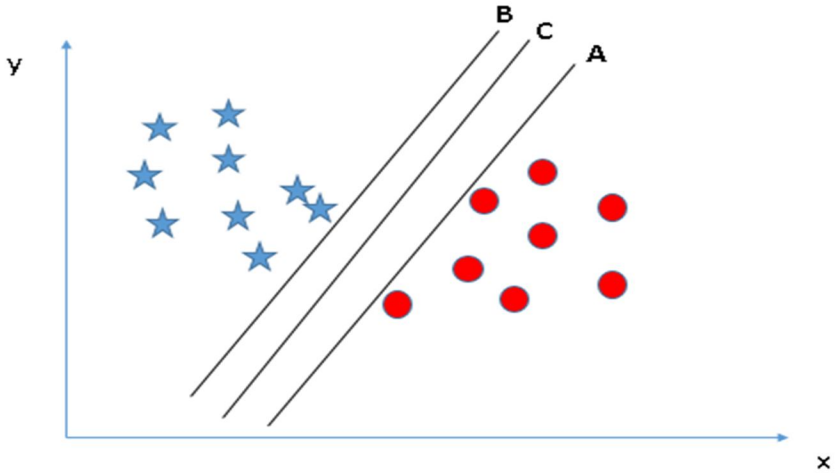


Fig. 5 Support vector machine hyperplane scenario - 2

©IJRASET: All Rights are Reserved

2) *Logistic Regression:* Logistic Regression [31] is a regression model. It deals with problems with two classes i.e., the binary classification problems. For e.g.: malicious or not malicious, spam or not spam etc. Logistic Regression is a supervised classification algorithm. In regression scenarios, we set a threshold value and based on that threshold decisions or outputs are made. It uses a function called sigmoid function for classifications. Consider a scenario where we need to decide whether a website is spam or not. Two types of regressions can be used for this prediction i.e. either linear regression or logistic regression. If we use liner regression for prediction, a threshold is set and based on it output can be predicted. As an example consider if the actual output is spam, the predicted value is 0.2 and the threshold value is set to be 0.7. From these results we can say that the predicted value is less than the threshold and hence the data is classified or predicted as not spam. Thus linear regression is not suitable for this type of prediction or classification problems. So we go for logical regression. Here instead of a threshold the value ranges from 0 to 1. There are different types [31] of Logistic Regression namely:

a) *Binary Logistic Regression:* The target can have two possible solutions 0 or 1. Example: Spam or not, malignant or not, win or loss, pass or fail, etc...

b) *Multinomial Logistic Regression:* The target can have three or more outcomes that are not ordered. Example: Predicting which disease is spread most (Disease A, Disease B, Disease C).

c) *Ordinal Logistic Regression:* It deals with target having three or more outputs. Example: Movie rating from 1 to 5 or a test score can be categorized as very poor, poor, good, very good, and so on.

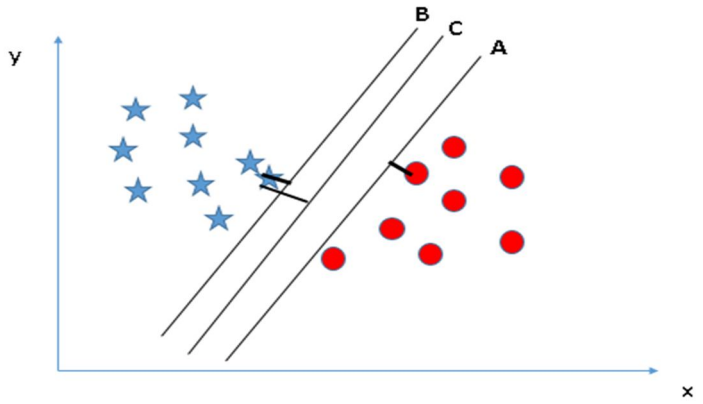


Fig. 6 Hyperplane with higher margin

3) *Decision Tree:* Decision Tree [32] is a supervised learning algorithm. It uses a tree like structure to represent data. By using that structure classifications can be done easier. There are leaf nodes and internal nodes. The leaf node represents the class labels or the final outputs. The internal nodes represent the attributes [21]. There are a number of nodes in the tree. Each node represents some test cases. It will have two answers i.e. either yes or no. The answers of the test case are represented in the edges. If yes it will move to next level nodes and this step continues until the leaf nodes. If 'no' the same step continues till the leaf nodes. It will provide the output by moving down the tree till the leaf node.. This process is repeated for every level of sub trees. The algorithm used in decision trees is known as the ID3 algorithm [33]. An example of a decision tree is illustrated in figure 7. It explains whether to play Badminton on a particular day or not. From that figure if the weather is sunny and humidity is normal then badminton can be played on that particular day. Similarly if the weather is sunny and humidity is high then badminton cannot be played. A decision tree will represent all the possible outputs.

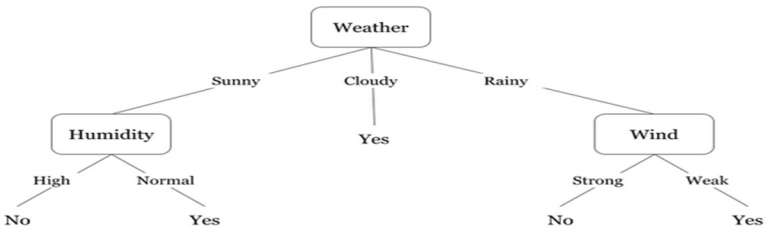


Fig. 7 Decision Tree illustrating whether to play badminton on a particular day or not

©IJRASET: All Rights are Reserved

B. Online Learning Algorithm

Online Learning [34], [35] uses a stream of data for classification i.e. it trains in successive rounds. The data are fed in a sequential and timely manner. At time‘t’ a stream of data are fed to the model and it will perform some prediction. The online learning algorithm will check whether the predicted value is correct or not. If the result is correct then it is fed back into the model. This process is repeated in further rounds. Figure 8 gives an illustration of Online Learning. When the datas are very large and the entire data cannot be fed into the model. In such cases online learning is used. It will split the data into various segments and fed into model in a sequential manner.

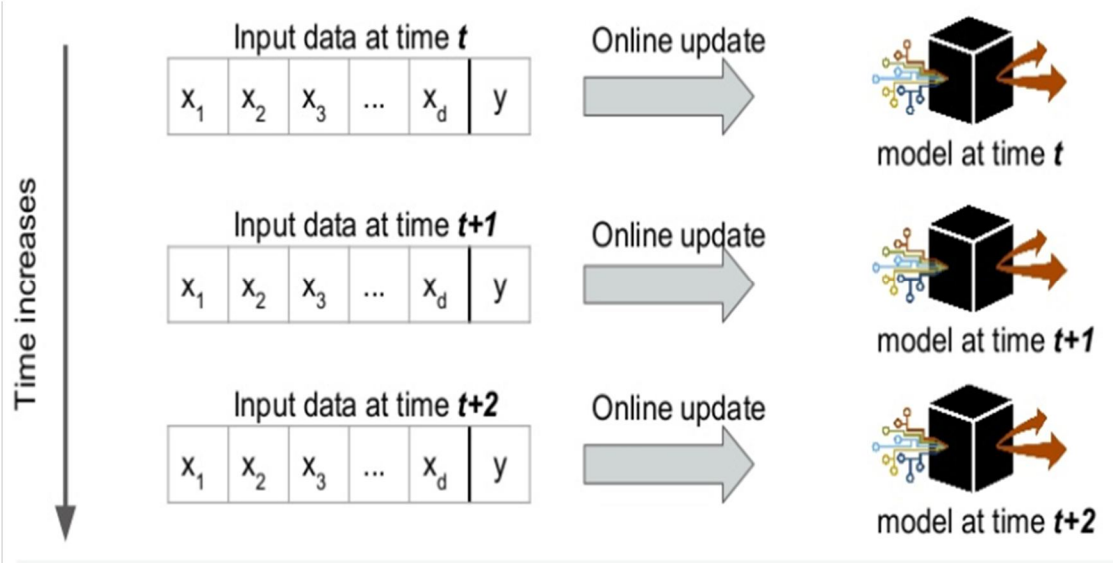


Fig. 8 Illustration of Online Learning

V. CONCLUSIONS

Malicious URL detection plays an important role in many cyber security applications. We perform a survey on different machine learning techniques and algorithms for Malicious URL Detection. It extracts the features of the URLs, learn a prediction model to classify a URL as malicious or benign. We also discussed various methods that can be used for the detection of malicious URLs, Various feature representations, and finally discusses various learning algorithms for resolving the malicious URL detection tasks.

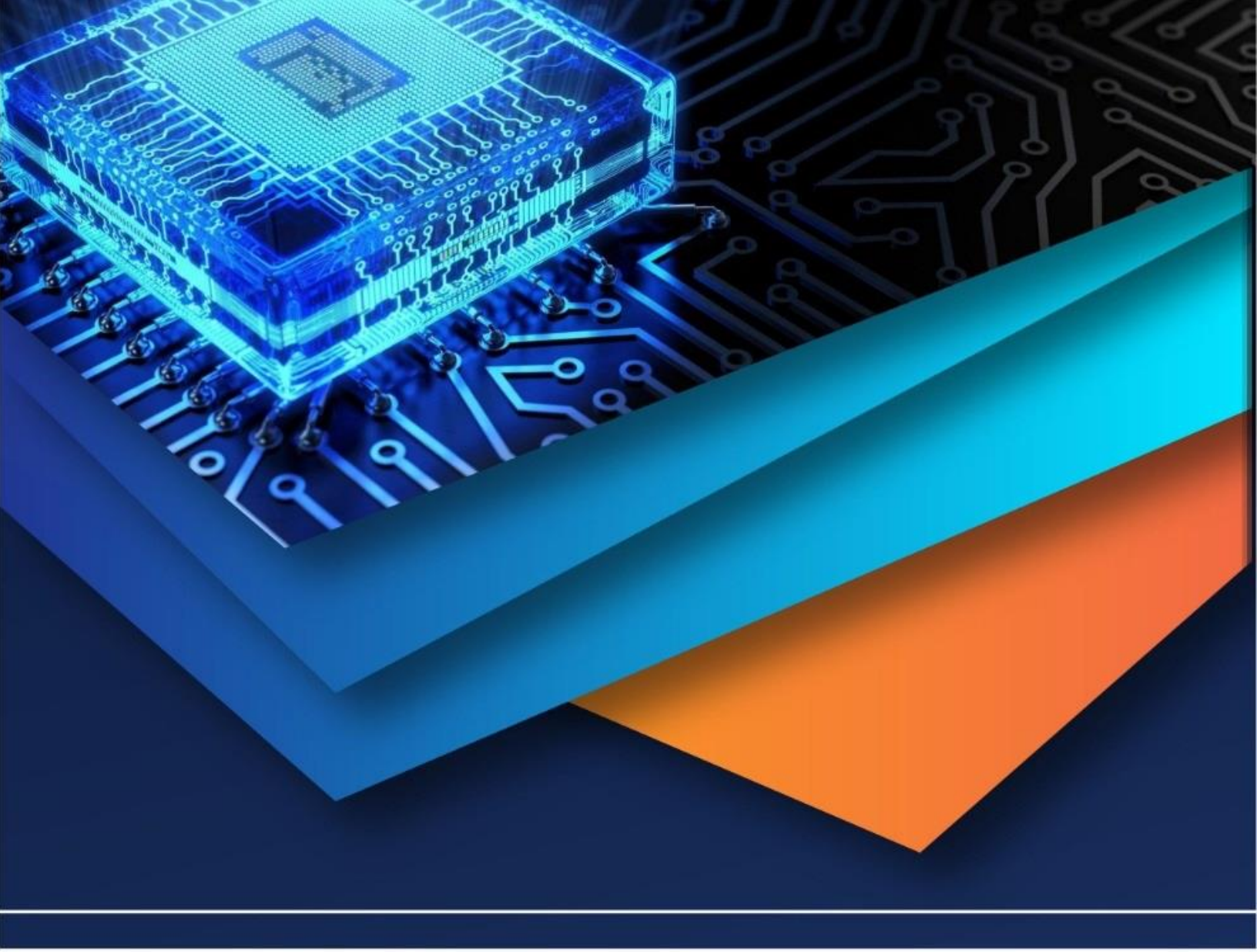
REFERENCES.

[1] Marco Cova, Christopher Kruegel, and Giovanni Vigna, “Detection and analysis of drive-by-download attacks and malicious JavaScript code,” In WWW. ACM,2010.
[2] Sushant Sinha, Michael Bailey, and Farnam Jahanian, “Shades of Grey: On the effectiveness of reputation-based blacklists," In MALWARE 2008. IEEE.
[3] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin, “A framework for detection and measurement of phishing attacks,” In Proceedings of the 2007 ACM workshop on Recurring malcode.
[4] Byung-Ik Kim, Chae-Tae Im, and Hyun-Chul Jung, “Suspicious malicious web site detection with strength analysis of a javascript obfuscation,” International Journal of Advanced Science and Technology (2011).
[5] Christoph Kolbitsch, Benjamin Livshits, Benjamin Zorn, and Christian Seifert, “Rozzle: De-cloaking internet malware,” In Security and Privacy (SP), 2012 IEEE Symposium on.
[6] Alexander Moshchuk, Tanya Bragin, Damien Deville, Steven D Gribble, and HenryMLevy, “SpyProxy: Execution based Detection of Malicious Web Content,” In USENIX Security.2007.
[7] Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel, “Prophiler: a fast filter for the large-scale detection of malicious web pages,” In Proceedings of the 20th international conference on World wide web. ACM.
[8] Birhanu Eshete, Adolfo Villafiorita, and Komminist Weldemariam, “Holistic analysis and detection of malicious web pages,” In Security and Privacy in Communication Networks. Springer.
[9] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones, “Phishing detection: a literature survey,” IEEE Communications SurveysTutorials (2013).
[10] Masahiro Kuyama, Yoshio Kakizaki, and Ryoichi Sasaki, “Method for Detecting a Malicious Domain by using WHOIS and DNS features,” In The Third International Conference on Digital Security and Forensics (DigitalSec2016).
[11] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker, “ Beyond blacklists: learning to detect malicious web sites from suspicious URLs,” In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,2009.
[12] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker, “Learning to detect malicious urls,” ACM Transactions on Intelligent Systems and Technology (TIST) (2011).
[13] D Kevin McGrath and Minaxi Gupta, “Behind Phishing: An Examination of Phisher Modification,” LEET(2008).

©IJRASET: All Rights are Reserved

[14] SUN Bo, Mitsuaki Akiyama, YAGI Takeshi, and Mitsuhiro Hatada, "Automating URL blacklist generation with similarity search approach," IEICE TRANSACTIONS on Information and Systems (2016).
[15] Mahmoud T Qassrawi and Hongli Zhang, "Detecting malicious web servers with honeyclients," Journal of Networks (2011).
[16] Konrad Rieck, Tammo Krueger, and Andreas Dewald, "Cujo: efficient detection and prevention of drive-by download attacks," In Proceedings of the 26th Annual Computer Security Applications Conference. ACM.2010.
[17] Pranam Kolari, Tim Finin, and Anupam Joshi, "SVMs for the Blogosphere: Blog Identification and Splog Detection," In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs.2006.
[18] Anh Le, Athina Markopoulou, and Michalis Faloutsos, "Phishdef: Url names say it all," In INFOCOM, 2011 Proceedings IEEE.
[19] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.2009.
[20] Enrico Sorio, Alberto Bartoli, and Eric Medvet. 2013, "Detection of hidden fraudulent URLs within trusted sites using lexical features," In Availability, Reliability, and Security (ARES), 2013 Eighth International Conference on. IEEE.
[21] Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," In Proceedings of the 20th international conference on World wide web. ACM.2011.
[22] Yung-Tsung Hou, Yimeng Chang, Tsuhan Chen, Chi-Sung Laih, and Chia-Mei Chen, "Malicious web content detection by machine learning," Expert Systems with Applications (2010).
[23] Guang Xiang, Jason Hong, Carolyn P Rose, and Lorrie Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security (TISSEC) (2011).
[24] Anthony Y Fu, Liu Wenyin, and Xiaotie Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)," IEEE transactions on dependable and secure computing (2006).
[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385 (2015).
[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural Networks," In Advances in neural information processing systems.2012.
[27] Yazan Alshboul, Raj Nepali, and Yong Wang, "Detecting malicious short URLs on Twitter," (2015).
[28] Sushma Nagesh Bannur, Lawrence K Saul, and Stefan Savage, "Judging a site by its content: learning the textual, structural, and visual features of malicious web pages," In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence. ACM.2011.
[29] Weibo Chu, Bin B Zhu, Feng Xue, Xiaohong Guan, and Zhongmin Cai, "Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs," In Communications (ICC), 2013 IEEE International Conference on. IEEE.
[30] Mingxing He, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, and Adi Sutanto, "An efficient phishing webpage detector," Expert Systems with Applications (2011).
[31] Sangho Lee and Jong Kim, "WarningBird: Detecting Suspicious URLs in Twitter Stream," In NDSS.2012.
[32] Anupama Aggarwal, Ashwin Rajadesingan, and Ponnurangam Kumaraguru, "Phishari: automatic real-time phishing detection on twitter," In eCrime Researchers Summit (eCrime), 2012. IEEE.
[33] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," In NDSS.2011
[34] Nicolo Cesa-Bianchi and Gabor Lugosi, "Prediction, learning, and games," Cambridge University Press.2006.
[35] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao, "Online Learning: A Comprehensive Survey," arXiv preprint arXiv:1802.02871 (2018).
[36] Danny Sullivan, "Google: 100 Billion Searches Per Month, Search To Integrate Gmail, Launching Enhanced Search App For iOS," Search Engine Land. http://searchengineland. com/google-search-press-129925 (2012).
[37] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker, "Identifying suspicious URLs: an application of large-scale online learning," In Proceedings of the 26th Annual International Conference on Machine Learning.2009.

©IJRASET: All Rights are Reserved



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)