Intelligent Malicious URL Detection with Feature Analysis

Yu-Chen Chen
Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
M10707503@mail.ntust.edu.tw

Yi-Wei Ma
Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
ywma@mail.ntust.edu.tw

Jiann-Liang Chen
Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
Lchen@mail.ntust.edu.tw

***Abstract*—The website security is an important issue that must be pursued to protect Internet users. Traditionally, blacklists of malicious websites are maintained, but they do not help in the detection of new malicious websites. This work proposes a machine learning architecture for intelligent detecting malicious URLs. Forty-one features of malicious URLs are extracted from the data processes of domain, Alexa and obfuscation. ANOVA (Analysis of Variance) test and XGBoost (eXtreme Gradient Boosting) algorithm are used to identify the 17 most important features. Finally, dataset is used to learn the XGBoost classifier, which has a detection accuracy of more than 99%.**

Keywords—malicious URL, JavaScript detection, artificial intelligence, feature analysis

I. INTRODUCTION

Hackers often use hot trend keywords and videos to distribute malicious programs or links to phishing websites that act maliciously on users' computers or defraud them by obtaining personal basic information from Internet [1]. Most of hacking attacks involve malicious websites to bait victims or software exploits, such as social email malicious attacks, SQL Injection, Distributed Denial-of-Service (DDoS) attacks or direct intrusion servers. Attacks against information security are diverse. Improving the awareness and protection of information security can effectively improve security of information.

Information security has three elements- confidentiality, integrity and availability [2]. Information security is required for Internet system services, Internet devices and the Internet of Things. Of these, Internet system services are most often used in attacks, involving drive-by downloads [3], buffer overflow, phishing websites, DDoS and SQL injection. Figure 1 presents the drive-by downloads attack. When a user browses a malicious website, a program on the website looks for exploits in the user's system and then tries to attack. If the attack is successful, the terminal device automatically downloads and executes the malware program or virus. At this point, the user's device becomes a member of the hacker's botnet.

Many studies with artificial intelligence techniques to detect the malicious URLs have been published recently [4-8]. The related studies are dedicated to different datasets and different intelligent approaches to malicious website detection, such as exploit different malicious website features, feature selection techniques, machine learning algorithms, neural network-like architectures, and network traffic-based concept drifts method. In this study, we roughly classify these studies on the detection of malicious websites into four approaches: web-based network traffic, URL keywords, web host information, and web content. In this study, based on web host information, web content features, and using machine learning to detect and protect against malicious URLs. It improves the disadvantage of blacklists [9] that is determining more unknown information and finding more malicious URLs.

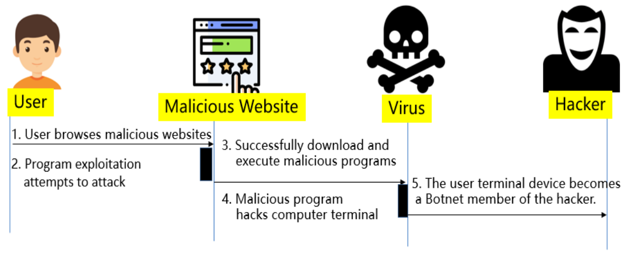


Fig 1. Drive-by Download Flow

Problem Statement

The main problem tackled in this study is to detect malicious URL in the benign URLs and provide 41 type features to an analyst, based on domain-based, Alexa-based, and obfuscation technique-based features from the Internet.

Approach and Contributions

The contribution in this work is the development of a malicious URL detection system and provide 41 feature, that includes three type features, one type is domain-based features, another is Alexa-based features, and the other is obfuscation technique-based features. According to features analysis, the F8, F4, and F5 are more important than other features. That can confirm our proposed feature is useful in this task. The performance of accuracy and precision can reach 99% and 100%.

II. FEATURE EXTRACTION

A dataset of benign website URLs and malicious website URLs is used in the experiments in this work. A Python web crawler and relevant open source programs are used to collect 41 domain-based, Alexa-based and obfuscation technique-based features. Original string of data is converted into numeric values for classification. Raw data are consolidated as shown in Figure 2. Various machine learning methods were used to find the maximally accurate classification, and to define useful features.

Length_Co	String_Fun	String_Rate	Dynamic_C	Unicode_C	Hex_Octal	Wrap_Cou	Space_Cou	Space_Rate	Domain_N	Org	DeCreation	DeUpdate_	DeExpiratic
0	0	0	0	0	0	0	0	0	0	0	10143	314	447
0	0	0	0	0	0	0	0	0	0	1	1415	281	45
0	0	0	0	0	0	0	0	0	0	0	670	326	59
315	0	0	0	0	0	0	25	7.94	0	1	6838	302	101
0	0	0	0	0	0	0	0	0	0	0	7581	281	87
0	0	0	0	0	0	0	0	0	0	0	1569	107	256
0	0	0	0	0	0	0	0	0	0	1	3087	171	199
0	0	0	0	0	0	0	0	0	0	0	1851	112	339
0	0	0	0	0	0	0	0	0	0	0	4156	98	3514
0	0	0	0	0	0	0	0	0	0	0	1038	306	422
378	0	0	0	0	0	0	34	8.99	0	0	5795	1478	779
0	0	0	0	0	0	0	0	0	0	0	5536	65	307
1787	0	0	0	0	0	0	224	12.53	0	0	9095	334	34
312291	3	0.08	16	0	1	0	111513	35.71	0	0	8533	875	1694
0	0	0	0	0	0	0	0	0	0	0	4532	179	946
0	0	0	0	0	0	0	0	0	0	0	4187	566	2387
0	0	0	0	0	0	0	0	0	0	0	7994	342	39
0	0	0	0	0	0	0	0	0	0	0	3430	70	221

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

Fig 2. Row Data

A. Data Collection

The dataset in this study consists of benign URLs and malicious URLs. The benign URLs which are network service system provided general organizations. The top five million sites were obtained from Alexa [10] and 13,027 unique benign URLs were selected. Malicious URLs were collected from open source datasets, such as the urlquery.net [11], urlscan.io [12] and GitHub [13], among others. In those malicious dataset needs query website index to collect URL via Virus Total, to confirm that is a malicious behavior URL. Accordingly, 13,027 unique malicious URLs were collected to ensure that the dataset was balanced. As a consequence, the dataset collected a total of 26,054 URLs, half of which were benign and half of which were malicious.

In order to understand the distribution of the two types of data in the dataset, using Auto Encoder-Decoder compresses the input vector according to the custom dense and then decompresses the output vector with the opposite dense, calculates the prediction error between the output and the input vector, and gradually improves the accuracy by using the back-propagation algorithm if the vector input trained Auto Encoder-Decoder model will the first encoder the vector, and the resulting middle layer cell is the essence of the input vector. The aim is to train a neural network for downscaling, while the data after downscaling is able to reconstruct the original data very well. During the training process, the difference between the output layer and the original amount of information is calculated, which is called the loss function (Loss), which is mathematically formulated as Equation (1). ($\widehat{x_i}$ is the output value; x_i is the input value; L is the loss function)

$$L(f(X)) = \frac{1}{2}\sum_{i=1}^{N}(\widehat{x_i} - x_i)^2 \qquad (1)$$

Using Auto Encoder to compress and map 41-dimensional features into 3-dimensional space is shown as Figure 3. In the 3-dimensional space diagram, which can notice that the samples almost overlap but calculated the top and bottom distances of the red and blue scatter diagram separately, and there is an error in the middle of the two categories. Also, it is known that malicious samples are broader than benign samples and have more diverse elements.

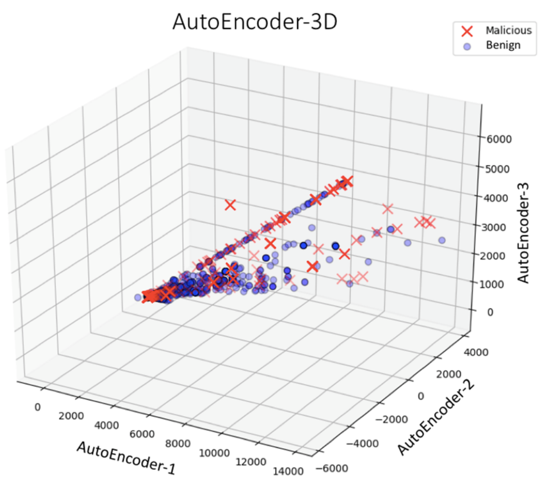


Fig 3. AutoEncoder-3D

B. Feature Information

The Python web crawler and open source programs were used to collect various features of URLs in the dataset. 41 features are obtained from domain-based, Alexa-based, and obfuscation technique-based. Data statistics are saved as a comma-separated CSV file. Some features, such as the domain, Org, ASN and others, are in a string format. Therefore, a method for converting a definition string into a numeric value is proposed, which will introduce the following feature tables.

In addition, the most significant contribution of this research is the feature table. We totally proposed 41 dimensions feature is shown as Tables 1, those are divided into three types, one is domain-based, another is Alexa-based features, the other is obfuscation technique-based features are shown as Tables 1,2, and 3. What follows is a description of 3 types of features tables.

Table 1: 41 Features

41 Features			
F1	*Domain*	F22	*Day7 PerUser*
F2	*Org*	F23	*Day1 PerUser*
F3	*Creation Time*	F24	*Comment Raws*
F4	*Update Time*	F25	*UnComment%*
F5	*DeExpiration*	F26	*Rediration*
F6	*Count DNS*	F27	*LinksInCount*
F7	*ASN*	F28	*Keyword Eval*
F8	*Country*	F29	*Avg String*
F9	*Count Trans*	F30	*Var Number*
F10	*Count Secure*	F31	*Plus Number*
F11	*Count IPv6*	F32	*Long*
F12	*Count Domains*	F33	*Wrap Number*
F13	*Count_IPs*	F34	*String_Number*
F14	*Count Countries*	F35	*Unicode Number*
F15	*Count Unpacked*	F36	*Hex Number*
F16	*Month3 Rank*		*Octal Number*
F17	*Month1 Rank*	F37	*Comment Number*
F18	*Day7 Rank*	F38	*Comment%*
F19	*Day1 Rank*	F39	*Document Location*
F20	*Month3 PerUser*	F40	*Eval Count*
F21	*Month1 PerUser*	F41	*Row Script*

1) Domain-based Features: The Domain Name System (DNS) is a service on the Internet that provides a decentralized database of domain names and IP addresses, allowing users to access network. WHOIS queries information about domain names, IPs, and owner's transmission protocol on the Internet. WHOIS users generally enter the domain name to be queried using the Command Line to obtain information from the WHOIS server. This feature type uses WHOIS query DNS-related information functions to extract the 15 features in Table 2.

Table 2: Domain-based Features

Feature	Description
Domain	Top ten common normal domain names (google.com,youtube.com,facebook.com,baidu.com ,wikipedia.org,yahoo.com,qq.com,taobao.com,gmail.com and twitter.com).If the domain is in the top ten domain names, then set to 1, otherwise, it is 0
Org	The maximum part of a normal domain is the same as Org name. Therefore, This feature is compared with Org and the DNS. If is the same string, then is 1; otherwise, it is 0

Creation Time	The amount of time between the creation of the domain and now
Update Time	The time between update of domain and now
Expiration Time	The amount of time between contract expiration of domain and now
Count DNS	The number of DNS
ASN	Top five common normal ASN and Org names(16509：Amazon, 203220：Yahoo, 32934：Facebook, 15169：Google and 11344：YouTube). If the ASN is in the top five ASN and Org, then set to 1, otherwise, it is 0
Country	If country is included in the top eleven common malicious country code(CN, US, EU, TR, RU, TW, BR, RO, IN, IT and HU), then is 1 ; otherwise, it is 0
Count Trans	Count HTTPs that are executed from DNS
Count Secure	Count HTTPs and IPs that are executed from DNS
Count IPv6	Count number of ipv6
Count Domains	Count domains from DNS
Count IPs	Count IPs from DNS
Count Countries	Count countries from DNS
Count Unpacked	The size of website at URL (in bytes)

2) Alexa-based Features: Alexa provides services for Amazon. It organizes the behaviors of users on the internet using big data, and monitors the traffic of all domains on the Internet. The Alexa website presents the global, national, and regional rankings of each website. Since benign links tend to be ranked high, malicious links are lower, the Alexa rank is used to extract eight features, in Table 3.

Table 3: Alexa-based Features

Feature	Description
Month3 Rank	Three-month website popularity ranking
Month1 Rank	Monthly website popularity ranking
Day7 Rank	Weekly website popularity ranking
Day1 Rank	Daily website popularity ranking
Month3 PerUser	Average number of monthly visits over three months
Month1 PerUser	Average number of daily visits in a month
Day7 PerUser	Average number of daily visits in a week
Day1 PerUser	Number of visits in a day

3) Obfuscation Technique-based Features: The obfuscation technique is an attack technique. It is commonly used by malicious websites to convert human-readable code into illegible code that cannot be read or understood. The purpose is to hide malicious code. Confusion technique can be achieved by many methods, through related papers [14-16] to propose the most common types of methods: Randomization Obfuscation, Code Obfuscation, and Encoding Obfuscation. The Obfuscation Technique is used herein to extract the 14 features on JavaScript that are shown in Table 4.

Table 4: Obfuscation Technique-based Features

Feature	Description
Comment Raws	Average number of comment per line in JavaScript
UnComment%	Percent rate of no comment program in JavaScript
Rediration Number	Number of redirect program in JavaScript
LinksInCount	Number of website links
Keyword Eval	Number of keywords, such as eval(), document.write(), etc. that programs frequently use for Obfuscation Technique in JavaScript
Avg String	Average number of string functions in JavaScript
Var number	Number of Var declarations in JavaScript
Plus number	Number of '+' operators in JavaScript
Long	Size of script in JavaScript
Wrap Number	Number of program newlines in JavaScript
String Number	Number of string functions in JavaScript
Unicode Number	Number of Unicode function in JavaScript
Hex Number and Octal Number	Number of Hex function in JavaScript
	Number of Octal function in JavaScript
Comment Number	Number of comment programs in JavaScript
Comment%	Number of comment programs as percentage in JavaScript
Document Location	Number of document function in JavaScript
Eval Count	Number of eval function in JavaScript
Row Script	Number of row function in JavaScript

III. PROPOSED DETECTION MECHANISM

A. Data Preprocess

Machine learning involves adjusting model weights and features of training data. Feature selection is an important process in this study. Removing redundant noise of Domain-based features, Obfuscation-based features, and Alexa-based features. Using ANOVA and XGBoost importance to reduce the complexity of the training model and reduce the overall model training time. In this work, 41 original features of the dataset are used. After analysis of the results from ANOVA and XGBoost, the number of features was reduced to 17, which are shown as Figure 4 and Figure 5. Table 5 shows the Top 17 features, obtained by an XGBoost comprehensive analysis of both the feature selection function and feature importance ranking.

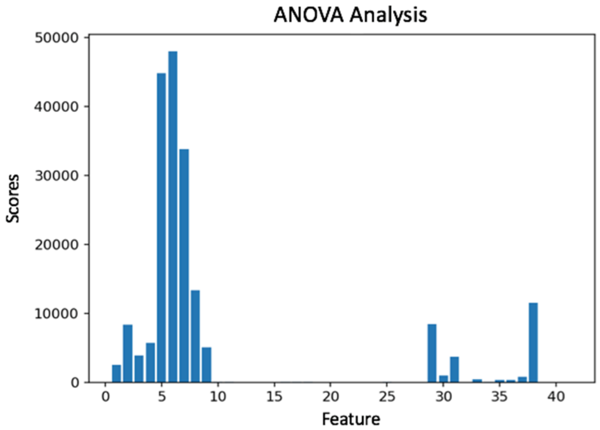


Fig 4. ANOVA Feature Selection

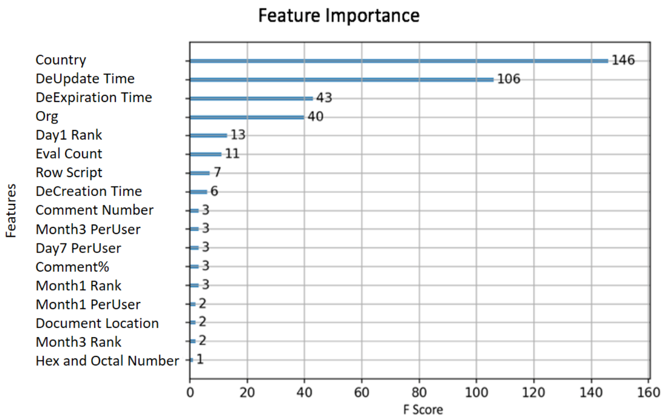


Fig 5. XGBoost Feature Importance

Table 5: FEATURE INTEGRATION

Top 17 Features		
Country	***Row Script***	***Month1 Rank***
DeUpdate Time	***DeCreation Time***	***Month1 PerUser***
DeExpiration Time	***Comment Number***	***Document Location***
Org	***Month3 PerUser***	***Month3 Rank***
Day1 Rank	***Day7 PerUser***	***Hex and Octal Number***
Eval Count	***Comment%***	

B. Machine Learning Mechanism

XGBoost is based on the Gradient Boosting Decision Tree (GBDT), which involves boosting technique of ensemble learning to reduce classified error margin worth [17]. Then, adjust the weight of the misclassified data features to learns what the error is, improving the results of the XGBoost classification.

XGBoost generalize loss values from square loss to a second-order deductible loss. The goal is to teach XGBoost model the value f to predict values of the form $f(x)$ during training. A T-leaf tree classifies data, and using a Taylor expansion of the loss function up to second order, which represents the smallest error values. Whenever a new tree is generated by fitting, view all of the generated trees, and selected the tree with the smallest objective function (cost), which represents the smallest error value, as shown in Eq. (2):

$$L^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2}$$

Where $g_i = \partial_{y^{(t-1)}} l(y_i, y^{(t-1)})$ and $h_i = \partial^2_{y^{(t-1)}} l(y_i, y^{(t-1)})$

The objective function is used to evaluate the fitness of a tree. To find the best segmentation point, the root node must be divided into two leaf nodes, based on the highest Information Gain of feature, which shown in Eq. (3):

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{3}$$

IV. 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 6. 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.

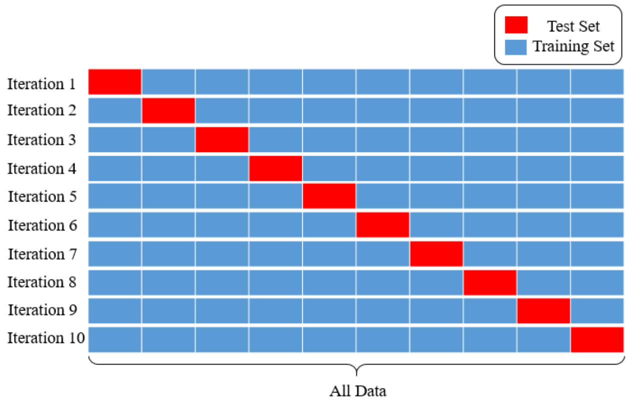
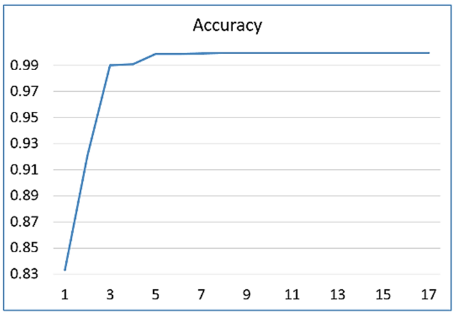


Fig 6. 10-fold Cross-Validation

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

V. CONCLUSIONS

This investigation proposes a machine learning architecture for detecting malicious URLs using the XGBoost algorithm. It generates a table of 41 kinds of malicious URL feature. Then, the accuracy of XGBoost classification model using the original 41 features is compared with that using the 17 most important extracted features. According the accuracy of 1 to 17 most important extracted features, the best number of features is the most important 1 to 9, which reduces the complexity of XGBoost classification model by 78%, increasing the training speed, while maintaining an accuracy of 99.98%.

REFERENCES

[1] Y.H. Chen and J.L. Chen, "AI@ntiPhish- Machine Learning Mechanisms for Cyber-Phishing Attack," IEICE Transactions on Information and Systems, vol. E102-D, no.5, pp. 878–887, 2019.

[2] F. Alkhudhayr, S. Alfarraj, B. Aljameeli and S. Elkhdiri, "Information Security: A Review of Information Security Issues and Techniques," 2nd International Conference on Computer Applications & Information Security, pp. 1–6, 2019.

[3] M. Cova, C. Kruegel, and G. Vigna, "Detection and Analysis of Drive-by-Download Attacks and Malicious javascript Code," 19th International Conference on World Wide Web, pp. 281–290, 2010.

[4] D. Muyang, Y. Han, and L. Zhao. "A Heuristic Approach for Website Classification with Mixed Feature Extractors." 2018 IEEE 24th International Conference on Parallel and Distributed Systems, pp. 134–141, 2018.

[5] A. Bhagwat, S. Dalvi, K. Lodhi and U. Kulkarni, "An Implemention of a Mechanism for Malicious URLs Detection," 2019 6th International Conference on Computing for Sustainable Global Development, pp. 1008–1013, 2019.

[6] 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.

[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] 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.

[9] Y. Fukushima, Y. Hori, and K. Sakurai, "Proactive Blacklisting for Malicious Web Sites by Reputation Evaluation Based on Domain and IP Address Registration", 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 352–361, 2011.

[10] Amazon, "Alexa," https://www.alexa.com/topsites.

[11] "Urlquery," https://urlquery.net/search.

[12] GmbH, "Urlscan.io," https://urlscan.io/.

[13] Cozpii, "Malicious URL Detection Dataset," GitHub, https://github.com/cozpii/Malicious-URL-detection.

[14] W. Xu, F. Zhang and S. Zhu, "The Power of Obfuscation Techniques in Malicious JavaScript Code: A Measurement Study," 2012 7th International Conference on Malicious and Unwanted Software, pp. 9–16, 2012.

[15] M. Cova, C. Kruegel, and G. Vigna, "Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code," 19th International Conference on World wide web, pp. 281–290, 2010.

[16] P. Likarish, E. Jung and I. Jo, "Obfuscated Malicious Javascript Detection using Classification Techniques ", 2009 4th International Conference on Malicious and Unwanted Software, pp. 47–54, 2009.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.