Using a Machine Learning Model for Malicious URL Type Detection

Suet Ping Tung[1,2], Ka Yan Wong[2], Ievgeniia Kuzminykh[3,4][0000-0001-6917-4234], Taimur Bakhshi[5][0000-0003-4750-7864], Bogdan Ghita[1][0000-0002-1788-547X]

[1] University of Plymouth, Drake Circus, Plymouth, PL4 8AA UK
bogdan.ghita@plymouth.ac.uk
[2] HKU School of Professional and Continuing Education, Kowloon Bay, Kowloon, Hong Kong
20040675@learner.hkuspace.hku.hk,
ivy.wong@teacher.hkuspace.hku.hk
[3] King's College London, Strand, London, WC2R 2LS, UK
[4] Kharkov National University of Radio Electronics, 14 Nauki av., Kharkov, Ukraine
ievgeniia.kuzminykh@kcl.ac.uk
[5] FAST National University of Computer & Emerging Sciences, Lahore, Pakistan
taimur.bakhshi@nu.edu.pk

Abstract. The world wide web, beyond its benefits, has also become a major platform for online criminal activities. Traditional protection methods against malicious URLs, such as blacklisting, remain a valid alternative, but cannot detect unknown sites, hence new methods are being developed for automatic detection, using machine learning approaches. This paper strengthens the existing state of the art by proposing an alternative machine learning approach, that uses a set of 14 lexical and host-based features but focuses on the typical mechanisms employed by malicious URLs. The proposed method employs random forest and decision tree as core mechanisms and is evaluated on a combined benign and malicious URL dataset, which indicates an accuracy of over 97%.

Keywords: Malicious URL, Web Security, Machine Learning, Phishing, Spamming, Malware, Lexical Feature, Traffic.

1 Introduction

There are currently over 4.66 billion active Internet users in the world, who rely on it to obtain information, communicate, or to support their work or daily activities [1]. According to the same report, the average user spends almost 7 hours on Internet every day for a range of activities, from online shopping and searching for information to social networking and work. While, through the media reports and user education across most organisations, people are aware of various aspects of cybersecurity, the level of knowledge and proficiency in defending against possible attacks is relatively low for a typical user. One of the most common attack vectors are malicious URLs, due to their convenience and ability to disguise or integrate within typical browsing. This trend was confirmed by Google transparency report, which identified over 2 million phishing websites in 2020 [2]. The risks posed by accessing such sites vary from private information disclosure to installation of malicious software on the computer used by the victim. The underlying attacks also vary, including techniques such as phishing, spamming, or drive-by-download.

The initial approach from the research community was to propose a series of countermeasures revolving around blacklisting of malicious URL or identifying malicious hosts. The lists are very dynamic, actively maintained by several organizations and communities, aiming to keep an accurate record of the current threats. While this line of protection is effective, the concept is a reactive one, as blacklists do not identify unknown malicious URLs. Therefore, a more recent alternative approach has been to apply machine learning algorithms that use specific features as inputs in order to detect malicious URLs. Due to their predictive nature, such approaches are far better in dealing with unknown malicious URLs and report a prediction accuracy rate up to 90%. This paper aims to strengthen the machine learning efforts to detect malicious URLs by proposing a hybrid solution that consists of a machine learning model used for detection and support it by manual input in order to evaluate its effectiveness.

The remainder of the paper is organized as follows. Section 2 presents the related work, then section 3 provides an outline of the approach. The model architecture, programming components, data collection and pre-processing, and model prediction measurements are in section 4. The model detection result and comparisons are then discussed in section 5. Section 6 concludes the paper with a summary of its achievements and limitations, as well as with possible avenues for future work.

2 Related works

Due to its associated potential threat level, malicious URL detection received in recent years a significant amount of attention from the research community. The researchers have used various discriminants to identify the malicious URLs, including lexical features (string properties of the URL, the number of special characters, length of URL, etc.), host-based features (domain name and hostname, IP address, location, etc.), content features (derived from HTML and JavaScript), or link popularity features (ranking, popularity score, reputation).

Based on the type of detection employed, the core mechanisms can be categorized into two areas: non-machine learning approaches and machine learning approaches. Using this criterion, blacklisting is categorized as a non-machine learning method to identify malicious URLs. In this category the research studies proposed implementation of blacklisting based on different techniques such as reputation-based [3] real-time blackhole lists [4], or tracking the top-level domain names [5]. This approach, while effective for known threats, is inherently likely to generate false negatives because of new malicious sites appearing. In addition, to avoid domain name blacklisting, attackers may employ a domain generation algorithm (DGA) to evade blacklists by generating new malicious URLs. The only option to keep ahead of the curve is to design adaptive, intelligent detection techniques, which apply machine learning (ML) algorithms to identify both URLs from the blacklists and the unknown malicious URLs. A summary of studies is presented in Table 1 and identifies what URL features and classifier have been used for detecting the non-benign web pages.

The authors of [6] conducted a comprehensive and systematic survey on malicious URL detection using machine learning techniques. The survey pointed out that support vector machine (SVM) and lexical features are the most widely used machine learning algorithm and type of features respectively. Many studies, such as [7–10], focus on detecting only phishing malicious web pages since vast majority of the malicious links in internet created for phishing purposes [11]. In this context, [7] and [8] used lexical-based URL features only to identify malicious links, while [12] extended the feature extraction with JavaScript client code analysis to achieve a better detection rate.

Considering the type of attack rather than the method of detection, [13] provides an overview of recent phishing URL detection studies. The authors reviewed 13 studies between 2014 and 2019 in terms of algorithms used, performance metrics and proc and cons in the study. Same authors made another survey [9] on the datasets used by researchers about malicious input for feature extraction and training of models [9]. The analysis showed that most studies use imbalanced datasets as the number of phishing sites cannot be compared with that of legitimate URLs.

Other studies aimed to refine the machine learning classification results by combining additional techniques; along this line of research, [14] reduced the false negative rate by using classification based on association (CBA) algorithm. The authors proposed a mix of lexical and comprehensive content-based features, which led to a false negative rate of 1.35%, a significant improvement from the 7.57% in the study that used only lexical features [8]. The authors did not evaluate the complexity of the proposed model, but other works showed that the extraction of content-based features requires more time and creates more delays when analysing a web page when compared to lexical features since reading of source code of the page, search for suspicious functions, iframes, parsing of DOM model are time demanding.

The authors in [15] applied range of ML algorithms such as Logistic Regression, Stochastic Gradient Descent (SGD), Random Forest (RF), Support-Vector Machines (SVM), Naive Bayes, k-Nearest Neighbors (kNN), and Decision Tree (DT) to the dataset with malicious URLs to investigate the prediction accuracy. They extracted the features such as, domain and sub-domain names and suffix to distinguish malicious web site from benign. Their dataset included collection of URLs from different sources and consisted of malware, hidden fraudulent and block listed URLs. All models showed high prediction accuracy, however, random forest algorithm attained the highest F1 score and accuracy. Following up on comparative studies, [16] focused on the host-based features such as domain details, IP addresses and port number to detect malicious web pages. In their experiments, the authors evaluated the effectiveness of different classification algorithms; the tree-based algorithm called Gradient-boosted tree showed the best results with an overall accuracy of 96.9%. The authors in [17] also run a comparative analysis, where they evaluated 7 detection models based on various ML algorithms picked up from their literature review study. Amongst the tested algorithms, the CMU [18] and Endgame [19] models based on bidirectional gated recurrent unit (BGRU) and long short-term memory (LSTM) yielded the highest accuracy.

As part of the machine learning domain, deep learning algorithms are one of the promising areas, due to their ability to replicate more complex behaviour. However, while they are all very effective at learning the patterns exhibited by the targeted phenomena, the inherent issue of deep learning algorithms is their high computational demand, particularly when employing a higher number of variable with wider value ranges [20]. The extent of the computational complexity increase was investigated in [21], where different algorithms were tested in terms of CPU, GPU, and TPU architectures. The deep learning algorithms showed higher accuracy than some of their ML counterparts, but the time required for training and for making decision was 2-4 times higher. In terms of accuracy, the best results showed RF algorithms with 98.68%.

The authors in [22] focused on improving classifiers by applying linear and non-linear transformation. This allowed them to improve the performance of certain ML algorithms, such as k-NN, SVM and Multi-layer Perceptron (MLP). The classifiers were also evaluated in terms of time efficiency on the testing and training subsets. It should be noted that authors used a large set of features (64 inputs in total) that caused the long computational time for the algorithms. Another study [23] used an open-source dataset of malicious URLs [24] (also used by [15] to evaluate the performance of classifiers). The authors established that the Naive Bayes algorithm performed better than logistic regression and convolutional neural network (CNN), with an accuracy 86.25%.

Table 1. Summary of machine learning approach in the literature.

Year	Ref	Features	ML algorithm	Description
2021	[14]	Lexical, Content-based	CBA	Benign page VS Malicious page
2021	[17]	Lexical, Host-based	BGRU, CNN, LSTM	Detect malicious URLs, file paths and registry keys Social media text classification
2021	[15]	Lexical, Host-based	Logistic Regression, SGD, RF, SVM, Naive Bayes, kNN, and Decision Tree	Normal page VS Malicious page
2020	[22]	Lexical, Host-based, Reputation based	kNN, L-SVM, Linear Discriminant Analysis (LDA), Logistic Regression	Normal page VS Malicious page
2020	[23]	Lexical, Host-based,	CNN, Logistic Regression, Naïve Bayes	Normal page VS Malicious page
2020	[21]	Lexical	RF, Decision Tree, kNN, SVM, Logistic Regression) LDA, AdaBoost, Naive Bayes, Fast.ai and Keras-TensorFlow	Normal page VS Malicious page
2018	[16]	Host-based	Decision Tree, Gradient-boosted tree (GBT), L-SVM, Naive Bayes, Random Forests	Benign page VS Malicious page
2016	[26]	Lexical, Host-based	C4.5, Decision Tree	Benign VS Malicious
2013	[27]	Lexical, Link popularity	Random Forest, Naive Bayes, Logistic Regression, J48, SVM	Normal page VS Malicious page
2013	[28]	Lexical, Host-based	SVM	Benign VS Phishing
2011	[25]	Lexical, Link popularity, Host-based	SVM, RAkEL, kNN	Classify attack types by URL
2011	[29]	Lexical, Webpage Content and Host-Based	Random Forest, Naive Bayes, Logistic regression, J48	Normal page VS Malicious page

The authors in [25] advised that understanding the attack type of malicious URL is useful for the user to respond properly. However, recent research papers seem to focus exclusively on benign and malicious, or benign and phishing. this indicates a lack of research papers addressing multiclass classification to detect the attack type of malicious URLs.

Drawing on the limitations of the existing research, this study aims to investigate the ability of machine learning algorithms to classify different types of malicious URLs using a reduced and robust set of parameters.

3 Methodology

As highlighted by the previous section, machine learning techniques represent an effective approach to identify malicious URLs due to their capacity to formally parse the content of the URL and compare it against recognizable malicious patterns. Following on their success, we propose a model that uses Decision Tree and Random Forest classification algorithms for detecting malicious URLs. This is in line with the conclusions from the authors of [15, 21, 27] who pointed out that Decision Tree and Random Forest are indeed the most likely to improve the detection accuracy, as they are effective classifiers, dynamically learning and adapting to variable domains [30].

The architecture of the proposed model for malicious URLs detection is presented in Fig. 1 and includes the process of extracting features, training phase to the classifiers, as well as the core decision tree and random forest techniques for predicting whether the URL is malicious or benign, as well as the type of attack that may be included in the URL.

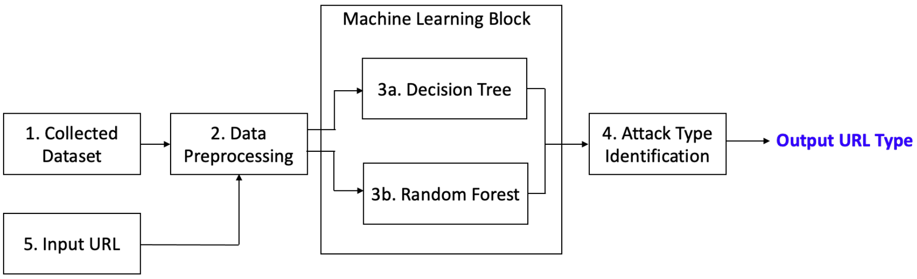

Fig. 1. The architecture of the model.

3.1 Datasets

As mentioned before, the aim of the proposed framework is to detect various types of URL attacks. In order to provide a robust knowledge base for training the model, we combined several existing datasets that included URLs of four types: benign, spam, phishing and malware. This led to four single-class datasets, resulting from the mixing of benign and malicious URLs matching a specific type of attack, and one multi-class dataset, including all four categories listed above. The final distribution of URLs in our dataset is presented in Fig. 2.

The dataset with benign URLs was taken from a study undertaken at University of New Brunswick [31]. It contains the URLs that have been collected from different Internet open source repositories, ordered by alexa.com [32]. From a half of million original URLs, the researchers extracted 35,300 that were labeled as benign after removing duplicates and virus checks.

For spam attacks, we used WEBSPAM-UK2007, a publicly available dataset collected by C. Castillo, supported by a team of volunteers [33]. The collection originally crawled 114,529 hosts of the .uk domain, and extracted 12,000 URLs that were labelled as spam web pages.

OpenPhish is one of the service providers that provides a blacklist of phishing URLs [34]. It contains millions of unfiltered URLs from a variety of sources and filter the web pages to detect phishing ones. We selected a subset of 10,000 URLs from this dataset.

DNS-BH is a project of RiskAnalytics that maintains list of domains that possible to spread malware and spyware [35]. A subset of 12,000 URLs, all related to malware websites, were obtained from this source.

In addition, in order to check the presence of a brand name in the domain name and path of each URL, we collected a subset of 50 sites from Alexa separately and added them to the benign dataset.

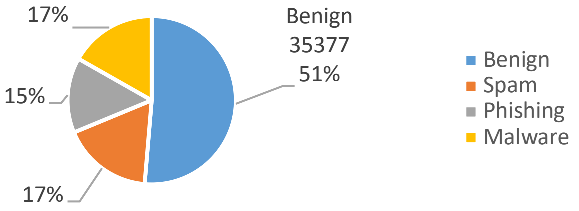

Fig. 2. Types of URLs in the dataset.

3.2 Dataset processing and features extraction

The effectiveness of a machine learning algorithm, beyond its ability to identify specific types of patterns, depends on the set of features used as input. The core idea behind our proposed approach is that malicious URLs tend to work either by redirecting the browser to a malicious page through obfuscation or path traversal or by loading an executable. We aim to identify these through an abnormal combination of non-alphanumeric characters such as slashes, equal signs, dashes, underscores, or dots, as well as extensions of executables, such as exe, bin, or configbin. As a result, we selected 14 features related to URL syntax, domain and path that belong to the lexical and host-based categories. Table 2 provides a summary of the features used in our model.

The novelty and benefits of this approach are two-fold: conceptual and efficiency. From a concept perspective, rather than the lexical or content-based approaches used by prior research, we focus on the typical mechanisms that URLs use for attack. From an efficiency perspective, the models we propose are rather lightweight, using 14 parameters as input rather than 50+ inputs, often including lexical parameters requiring a dictionary comparison.

Table 2. Selected URL features for the model.

Category	Name	Description
Lexical	url_length	The length of the URL
	sp_char_count	Total specific characters in URL
	slash_count	Number of slashes (/) in URL
	token_count	Number of tokens in URL
	equality_count	Number of equality (=) in URL
	dash_count	Number of dash (-) in URL
	underscore_count	Number of underscore (_) in URL
	dot_count	Number of dots (.) in URL
	exe_count	Number of .exe in URL
	bin_count	Number of .bin URL
	configbin_count	Number of .configbin in URL
Host-based	is_IP	Presence of IP address in domain name
	brandInSLD	Presence of brand name in domain name
	brandExist	Presence of brand name in path

The datasets listed in the previous section were pre-processed to ensure data consistency (URL format). The features were extracted using a parser and merged with the URL type in order to create the dataset.

3.3 Model Classification and Cross-Validation

Classification of data was done through two machine learning algorithms: Decision Tree [30] and Random Forest [36]. We used the cross-validation technique [37] to train the data. While this technique is typically used for small datasets, we preferred it because it allows to preserve a quality and sample size during splitting the dataset into training and testing sets. The basic approach in cross-validation is showed in Fig. 3 and performed as the data is split into k-folds: the training set is split into k smaller subsets, to average the computed score.

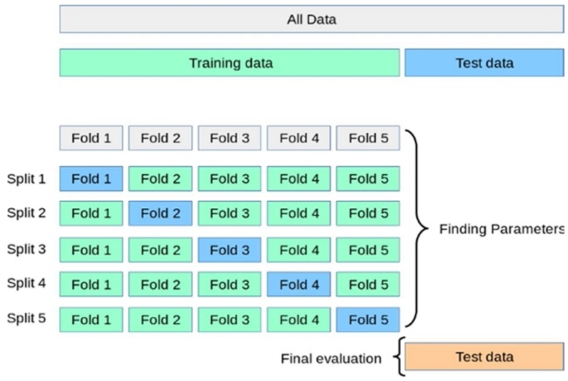

Fig. 3. Visual explanation of data splitting in k-fold cross-valuation [37]

The following procedure is applied to each of the k-folds: the model is trained using k-1 of the folds as training data and then the resulting model is validated on the remaining part of the data (i.e., it is used as a test set to compute a performance measure such as accuracy). The average of the values after each split is the final model performance.

The accuracy of the model was determined by calculating the ratios between the variables describing the outcome of the classification: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). We used three common metrics to evaluate the accuracy of the classifiers: precision, recall and F1-score.

Precision is a true positive predictive value of class, representing the ratio between the number of true positives (TP) and the total number of predicted positive class. Recall, same as sensitivity, is a metric for evaluating the correctness of the class and it is defined as the ratio between the number of true positives and the total number of predictions of the respective class. The F1-score is the harmonic mean of recall and precision. The distinctive feature of F1- score is defined both on the positive and negative classes and F1-score is average these two values. Table 3 below summarises the three indicators.

Table 3. Model accuracy metrics.

Metric	Definition
Precision	TP / (TP + FP)
Recall (sensitivity)	TP / (TP + FN)
F1-Score	2 x (Precision x Recall / Precision + Recall)

4 Results

We applied the k-fold cross-validation technique to the dataset and generated used a 30/70 ratio for the testing/training subsets. As shown in the breakdown from Table 4, out of the total 68951 URLs, the training subset included 48265 samples while the testing subset had 20686 samples. The ratio of benign and malicious samples was approximately 50-50 across the two subsets due to the random nature of selection, which also minimised a possible imbalance between training and testing.

Table 4. Training and testing dataset samples.

Type	Training data	Testing data	Total
Benign	24814	10612	35426
Phishing	6962	2999	9961
Spam	8284	3715	11999
Malware	8205	3360	11565
Total, URLs	48265	20686	68951
Total, %	70	30	100

Following the split, the dataset was fit to the two designed models - Decision Tree and Random Forest - and the resulting performance was evaluated using precision, recall and F1-score.

4.1 Decision Tree

The result of the decision tree model is summarized in Table 5. The final accuracy of Decision Tree model is 96.33% and the F1- score is over 90% of each attack type. The mean of the k-fold cross-validation score is 95.42%, which close to the final testing dataset accuracy.

Table 5. Decision Tree prediction result

Type	Precision	Recall	F1-score
Benign	0.987	0.988	0.987
Phishing	0.913	0.889	0.901
Spam	0.964	0.960	0.962
Malware	0.932	0.956	0.944
Accuracy: 96.33%			

4.2 Random Forest

The results of the Random Forest model are presented in Table 6. The final accuracy of Decision Tree model is 97.49% and F1-score is over 93% of each attack type. The mean of k-fold cross-validation score is 96.67%, which is very close to the final accuracy.

Table 6. Random Forest prediction result

Type	Precision	Recall	F1-score
Benign	0.988	0.995	0.991
Phishing	0.943	0.925	0.934
Spam	0.943	0.925	0.934
Malware	0.943	0.925	0.934
Accuracy: 97.49%			

4.3 Discussion

The performance results of the Decision Tree and Random Forest models showed that Random Forest model is more effective on detection of all the types of malicious URL, given that precision, recall and F1-score of each attack type are higher than Decision Tree, there are over 95% of all the types except phishing.

Our research result is very close to that of the work done by Choi et al. [25]. Although a major part of our experiment datasets (benign, phishing, malware, a portion of spam) are identical, we have extended our dataset with Defacement dataset. Regarding lexical classification outcomes of Choi et al. (Spam 73 %Phishing 91.6 % and Malware 70.3 %), authors did not mention precisely whether their result stems from applying multi-class or single-class classifier. Note that using multi-class classification with additional dataset must degrade the overall performance and accuracy. However, our Random Forest classifier outperforms their lexical feature results in either case of individual and aggregated (multi-class) classifiers yielding around 99 % and 97 % accuracy respectively even with an addition of Defacement URL dataset.

5 Conclusions

This study proposed a novel parameter set for detection of malicious URL, focused on discriminating various types of behaviour, using two machine learning algorithms, Decision Tree and Random Forest. The dataset for training was consolidated by using several existing datasets with benign and malicious URLs, then fed to both algorithms to predict the attack type of the URL. Selected feature sets applied on supervised classification on a ground truth dataset yields a classification accuracy of 97 % with a low false positive rate. Our prediction interval filtering experiment can also be helpful to improve classifier accuracy. In addition, it can be extended to calculate the risk rating of a malicious URL after parameter adjustment and learning with huge training data. The random forest classification accuracy marginally outperformed decision tree, as it was able to identify approximately 97 % of the malicious or benign URL.

For future work we are aiming to extend the work to a wider range of variable values, to reflect further, more complex malicious URL behaviour and ensure that the proposed methods remain up to date and continue to detect behaviour in the underlying web technology.

References

1. We Are Social, Hoootsuite: Digital 2021 Global Overview Report. Datareportal.com. 299 (2021).
2. Google: Google: Transparency Report. Google Transpar. Rep. (2010).
3. Prakash, P., Kumar, M., Rao Kompella, R., Gupta, M.: PhishNet: Predictive blacklisting to detect phishing attacks. Proc. - IEEE INFOCOM. (2010). https://doi.org/10.1109/INFCOM.2010.5462216.
4. Felegyhazi, M., Kreibich, C., Paxson, V.: On the potential of proactive domain blacklisting. LEET 2010 - 3rd USENIX Work. Large-Scale Exploit. Emergent Threat. Botnets, Spyware, Worms, More. (2010).
5. Sinha, S., Bailey, M., Jahanian, F.: Shades of Grey: On the effectiveness of reputation-based blacklists. 3rd Int. Conf. Malicious Unwanted Software, MALWARE 2008. 57–64 (2008). https://doi.org/10.1109/MALWARE.2008.4690858.
6. Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL Detection using Machine Learning: A Survey. (2017).
7. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based Associative Classification data mining. Expert Syst. Appl. 41, 5948–5959 (2014). https://doi.org/10.1016/j.eswa.2014.03.019.
8. Jeeva, S.C., Rajsingh, E.B.: Intelligent phishing url detection using association rule mining. Human-centric Comp. Inf. Sci. 6, (2016). https://doi.org/10.1186/s13673-016-0064-3.
9. Aung, E.S., Yamana, H.: URL-based phishing detection using the entropy of non- A lphanumeric characters. ACM Int. Conf. Proceeding Ser. (2019). https://doi.org/10.1145/3366030.3366064.
10. Ravi, R., Shillare, A.A., Bhoir, P.P., Charumathi, K.S.: URL based Email Phishing Detection Application. Int. Res. J. Eng. Technol. 8, 335–360 (2021).
11. Verizon: Data Breach Investigations Report (DBIR). Comput. Fraud Secur. 12, 8 (2019).

12. Hadi, W., Aburub, F., Alhawari, S.: A new fast associative classification algorithm for detecting phishing websites. Appl. Soft Comput. J. 48, 729–734 (2016). https://doi.org/10.1016/j.asoc.2016.08.005.
13. Aung, E.S., Zan, T., Yamana, H.: A Survey of URL-based Phishing Detection. 1–8 (2019).
14. Kumi, S., Lim, C., Lee, S.G.: Malicious url detection based on associative classification. Entropy. 23, 1–12 (2021). https://doi.org/10.3390/e23020182.
15. Shantanu, Janet, B., Joshua Arul Kumar, R.: Malicious URL Detection: A Comparative Study. Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021. 1147–1151 (2021). https://doi.org/10.1109/ICAIS50930.2021.9396014.
16. Tan, G., Zhang, P., Liu, Q., Liu, X., Zhu, C., Dou, F.: Adaptive Malicious URL Detection: Learning in the Presence of Concept Drifts. Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018. 737–743 (2018). https://doi.org/10.1109/TrustCom/BigDataSE.2018.00107.
17. Srinivasan, S., Vinayakumar, R., Arunachalam, A., Alazab, M., Soman, K.: DURLD: Malicious URL Detection Using Deep Learning-Based Character Level Representations. Malware Anal. Using Artif. Intell. Deep Learn. 535–554 (2021). https://doi.org/10.1007/978-3-030-62582-5_21.
18. Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W.: Tweet2Vec: Character-based distributed representations for social media. 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Short Pap. 269–274 (2016). https://doi.org/10.18653/v1/p16-2044.
19. Anderson, H.S., Woodbridge, J., Filar, B.: DeepDGA: Adversarially-tuned domain generation and detection. AISec 2016 - Proc. 2016 ACM Work. Artif. Intell. Secur. co-located with CCS 2016. 13–21 (2016). https://doi.org/10.1145/2996758.2996767.
20. Kuzminykh, I., Shevchuk, D., Shiaeles, S., Ghita, B.: Audio interval retrieval using convolutional neural networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 12525 LNCS, 229–240 (2020). https://doi.org/10.1007/978-3-030-65726-0_21.
21. Johnson, C., Khadka, B., Basnet, R.B., Doleck, T.: Towards detecting and classifying malicious urls using deep learning. J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl. 11, 31–48 (2020). https://doi.org/10.22667/JOWUA.2020.12.31.031.
22. Li, T., Kou, G., Peng, Y.: Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. Inf. Syst. 91, (2020). https://doi.org/10.1016/j.is.2020.101494.
23. Vundavalli, V., Barsha, F., Masum, M., Shahriar, H., Haddad, H.: Malicious URL Detection Using Supervised Machine Learning Techniques. ACM Int. Conf. Proceeding Ser. (2020). https://doi.org/10.1145/3433174.3433592.
24. Urcuqui, C.: Malicious and Benign Websites dataset, https://www.kaggle.com/xwolf12/malicious-and-benign-websites, last accessed 2021/07/12.
25. Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. WebApps. 11 (2011).
26. Mašetic, Z., Subasi, A., Azemovic, J.: Malicious Web Sites Detection using C4.5 Decision Tree. Southeast Eur. J. Soft Comput. 5, (2016). https://doi.org/10.21533/scjournal.v5i1.109.
27. Eshete, B., Villafiorita, A., Weldemariam, K., Zulkernine, M.: EINSPECT: Evolution-guided analysis and detection of malicious web pages. Proc. - Int. Comput. Softw. Appl. Conf. 375–380 (2013). https://doi.org/10.1109/COMPSAC.2013.63.

28. Chu, W., Zhu, B.B., Xue, F., Guan, X., Cai, Z.: Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. IEEE Int. Conf. Commun. 1990–1994 (2013). https://doi.org/10.1109/ICC.2013.6654816.
29. Canali, D., Cova, M., Vigna, G., Kruegel, C.: Prophiler: A fast filter for the large-scale detection of malicious web pages. Proc. 20th Int. Conf. World Wide Web, WWW 2011. 197–206 (2011). https://doi.org/10.1145/1963405.1963436.
30. S., M.: "Automatic construction of decision trees from data: A multidisciplinary survey. Data Min. Knowl. Discov. (1998).
31. Canadian Institute for Cybersecurity: URL dataset (ISCX-URL-2016).
32. Amazon: Alexa Internet, www.alexa.com.
33. Castillio, C.: Web Spam Collections, http://chato.cl/webspam/datasets/uk2007/, last accessed 2021/07/12.
34. OpenPhish: Phishing Intelligence. (2020).
35. Risk Analytics: DNS-BH - Malware Domain Blocklist. (2021).
36. Breiman, L.: Random Forests. Mach. Learn. 5–32 (2001). https://doi.org/10.1023/A:1010933404324.
37. Stone M.: Cross-Validatory Choice and Assessment of Statistical Predictions. J. R. Stat. Soc. Ser. B. 36, 111–147 (1974).