ISSN 2063-5346

SUSPICIOUS URL DETECTION USING MACHINE LEARNING APPROACHES

[1]Chandrakala B M, [2]B V Shruti, [3]Sarala D V

[1] Associate Professor, Department of Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore-560078.

[2]Associate Professor, Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bangalore- 560064.

[3] Assistant Professor, Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bangalore-560078.

*Corresponding Author: chandrakalabm-ise@dayanandasagar.edu

Article History: Received: 03.04.2023 **Revised:** 25.04.2023 **Accepted:** 15.05.2023

Abstract

Malicious URL, a.k.a. malicious site, is a typical and genuine danger to network safety. Vindictive URLs have spontaneous substance (spam, phishing, drive-by downloads, and so on) and bait clueless clients to become casualties of tricks (financial misfortune, burglary of private data, and malware establishment), and cause misfortunes of billions of dollars consistently. It is basic to identify and follow up on such dangers in a convenient way. Customarily, this recognition is done for the most part through the utilization of boycotts. Be that as it may, boycotts can't be thorough, and need the capacity to distinguish recently created malignant URLs. To improve the over-simplification of noxious URL locators, AI methods have been investigated with expanding consideration as of late. This article points

to give a thorough review and an underlying comprehension of Malicious URL Detection strategies

utilizing AI. We present the proper detailing of Malicious URL Detection as an AI task, and arrange and audit the commitments of writing contemplates that tends to various measurements of this issue (highlight portrayal, calculation plan, and so forth) Further, this article gives an opportune and exhaustive overview for a scope of various crowds, not just for AI specialists and engineers in scholarly community, yet in addition for experts and professionals in network safety industry, to help them comprehend the cutting edge and work with their own exploration and useful applications. We likewise examine functional issues in framework configuration, open exploration difficulties, and point out significant bearings for future research.

Keywords

Malicious URL Detection, Machine Learning, Online Learning, Internet security, Cyber security.

1.Introduction

Lately, data security has gotten a stylish subject since numerous individuals have experienced spillage of staff data. Simultaneously, assailants attempt to mimic as an approved individual or association. They utilize any type of medium to pull in clients, for example, adding influential promotions or pop-ups in informal organization administrations, installing counterfeit connections in messages or bargaining an real site. Such fakes are known as phishing. To be essentially characterized, phishing is a kind of danger of staff data where phishers deliberately assault an individual or on the other hand association. Inside three-years time frame. Additionally, it was accounted for that a little less than half of business email bargain (BEC) assaults used area names enrolled by the lawbreakers. They made comparative area names of confided in existing organization names to trap artless clients. 54% of BEC

assaults utilized free webmail in the Q3. Furthermore, around 66% of all phishing locales answered to APWG utilized SSL insurance, which was the most elevated rate since 2015, showing that clients can't totally depend on SSL. These reports guarantee that URLs have been a vector to be beguiled by phishers since regular clients are not completely mindful to dubious URLs. Our work expects to overview a differing pattern of malevolent URL location and to dissect an assortment of recognition methods changing after some time and one pack of conventional customer data. The exploratory outcomes show that our adversary of crawler methodology can effectively recognize every last one of those crawlers. URL is the shortening of Uniform Resource Locator, which is the worldwide location of archives what's more, different assets on the World Wide Web. A URL has two principle segments : (I) convention identifier (shows what convention to utilize) (ii) asset name (determines the IP address or the area name where the asset is found). The convention identifier and the asset name are isolated by a colon and two forward cuts, for example Figure 1.

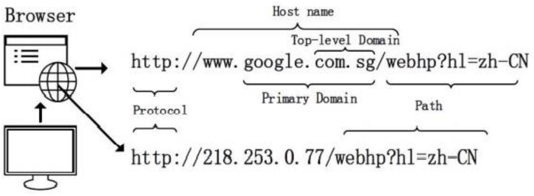

Browser
Host name
Top-level Domain
http://www.google.com.sg/webhp?hl=zh-CN
Protocol
Primary Domain
Path
http://218.253.0.77/webhp?hl=zh-CN

Fig. 1. Example of a URL - "Uniform Resource Locator"

Traded off URLs that are utilized for digital assaults are named as malevolent URLs. Indeed, it was noticed that near 33% of all sites are possibly pernicious in nature, illustrating uncontrolled utilization of vindictive URLs to execute digital wrongdoings. A Malicious URL or a noxious site has an assortment of spontaneous substance as spam, phishing, or drive-by download to dispatch assaults. Clueless clients visit such sites and become casualties of different kinds of tricks, counting financial misfortune, robbery of private data (character, charge cards, and so on), and malware establishment. Well known sorts of assaults utilizing malevolent URLs include: Drive-by Download, Phishing also, Social Engineering, and Spam. Drive-by download alludes to the (accidental)download of malware upon simply visiting a URL. Such assaults are typically completed by misusing weaknesses in modules or embeddings noxious code through JavaScript. Phishing and Social

Designing assaults stunt the clients into uncovering private or delicate data by imagining to be certifiable site pages. Spam is the utilization of spontaneous directives to promote or on the other hand phishing. These assaults happen in enormous numbers and have caused billions of dollars worth of harm, some in any event, abusing cataclysmic events. Powerful frameworks to identify such noxious URLs in an opportune way can extraordinarily assist with countering huge number of and an assortment of digital protection dangers. Thus, analysts and professionals have attempted to plan successful answers for Malignant URL Detection.

2. Background Work and Related work

In this overview, we audit the best in class machine learning methods for malevolent URL recognition in writing. We explicitly center around the commitments made for include portrayal furthermore, learning calculation improvement in this area. We efficiently order the different kinds of highlight portrayal utilized for making the preparation information for this assignment, and furthermore order different learning calculations used to gain proficiency with a decent forecast model. We additionally examine the open exploration issues and recognize bearings for future examination. We initially examine the general classifications of techniques utilized for distinguishing vindictive URLs - Blacklists (and Heuristics) and Machine Learning. We formalize the setting as an AI issue, where the essential necessity is acceptable element portrayal and the learning calculation utilized. We at that point exhaustively present different kinds of include portrayal utilized for this issue. This is trailed by introducing different calculations that have been utilized to tackle this errand, and have been created dependent on the properties of URL information. At long last we examine the recently arising idea of Malicious URL Detection as a help and the standards to be utilized while planning such a framework. We end the review by examining the pragmatic issues and open issues.

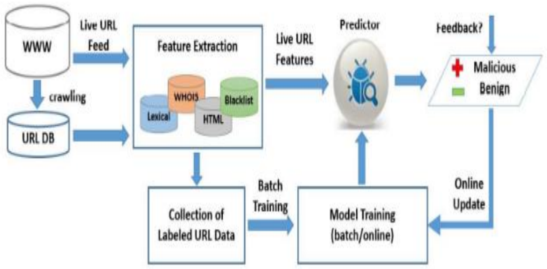


Fig. 2. A general processing framework for Malicious URL Detection using Machine Learning

3. MALICIOUS URL DETECTION

We first present the vital standards to address Malicious URL discovery, trailed by formalizing it as a ML Task.

3.1 Outline of Principles of Detecting Malicious URLs: An assortment of approaches have been

endeavored to handle the issue of Malicious

URL Detection. As indicated by the key standards, we classify them into: (I) Blacklisting or Heuristics, what's more, (ii) Machine Learning draws near

3.1.1 Heuristic Approaches: Boycotting approaches are a typical and traditional strategy for identifying noxious URLs, which frequently keep a rundown of URLs that are known to be pernicious. At whatever point another URL is visited, a data set query is performed. On the off chance that the URL is available

in the boycott, it is viewed as vindictive and afterward an admonition will be produced; else it is thought to be benevolent. Boycotting experiences the powerlessness to keep a comprehensive rundown of all

conceivable vindictive URLs, as new URLs can be effectively produced every day, along these lines making it unimaginable for them to recognize new dangers. This is especially of basic concern when the aggressors produce new URLs algorithmically, and would thus be able to sidestep all boycotts. Regardless of a few issues looked by boycotting, because of their straightforwardness and productivity, they keep on being one of the most usually utilized strategies by numerous enemy of infection frameworks today. Heuristic methodologies are a sort of augmentation of Blacklist techniques, wherein the thought is to make a

Interruption Detection Systems can check the website pages for such marks, and raise a banner if some dubious conduct is found. These techniques have preferred speculation abilities over boycotting, as they can recognize dangers in new URLs also. In any case, such techniques can be intended for just a set number of regular dangers, and can not sum up to a wide range of (novel) assaults. Additionally, utilizing obscurity strategies, it isn't hard to sidestep them. A more explicit form of heuristic methodologies is through investigation of execution elements of the site page Here additionally, the thought is to search for a mark of noxious action like uncommon interaction creation, rehashed redirection, and so on These techniques fundamentally require visiting the website page and consequently the URLs really can make an assault. Thus, such methods are frequently carried out in controlled climate like an expendable virtual machine. Such methods are very asset concentrated, and require all execution of the code (counting the rich customer sided code). Another disadvantage is that sites may not dispatch an assault right away subsequent to being visited, and consequently may go undetected.

3.1.2 Machine Learning Approaches : These methodologies attempt to examine the data of a URL and its comparing sites or site pages, by extricating great component portrayals of URLs, and preparing a forecast model on preparing information of both malevolent and favorable URLs. There are two-types

of highlights that can be utilized - static highlights, and dynamic highlights. In static investigation, we perform

the investigation of a website page dependent on data accessible without executing the URL (i.e., executing

JavaScript, or other code). The highlights separated incorporate lexical highlights from the URL string, data about the host, and now and again even HTML and JavaScript content. Since no execution is required, these techniques are more secure than the Dynamic methodologies. The basic supposition that will be that the dispersion of these highlights is diverse for pernicious and benevolent URLs. Utilizing this dispersion data, a forecast model can be fabricated, which can make expectations on new URLs. Because of the moderately more secure climate for extricating significant data, and the capacity to sum up to a wide range of dangers (not simply normal ones which must be characterized by a mark), static examination strategies have been broadly investigated by applying machine learning techniques.

3.1.3 Logistic Regression:

It is a statistical method for analysing a data set in a factual technique for examining an informational index where there are at least one free factors that decide a result. The result is estimated with a dichotomous variable (where there are just two potential results). The objective of strategic relapse is to track down the best fitting model to depict the connection between the dichotomous quality of premium (subordinate variable = reaction or result variable) and a bunch of autonomous (indicator or informative) factors. Strategic relapse is a Machine Learning order calculation that is utilized to foresee the likelihood of a straight out subordinate variable. In strategic relapse, the reliant variable is a paired variable that contains information coded as 1 (indeed, achievement, and so forth) or 0 (no, disappointment, and so on)

3.1.4 Support Vector Machine

It constructs grouping on relapse models as a tree structure. It separates an informational index into more modest and more modest subsets while simultaneously a related choice tree is steadily evolved. A classifier that categorizes the data set by setting an optimal hyper plane between data. I chose this classifier as it is incredibly versatile in the number of different kernelling functions that can be applied and this model can yield a high predictability rate. Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

3.1.5 Random forests:

Arbitrary choice timberlands are an outfit learning technique for arrangement, relapse and different assignments, that work by building a huge number of choice trees at preparing time and yielding the class that is the method of the classes (grouping) or mean forecast (relapse) of the individual trees. Irregular choice woodlands right for choice trees' propensity for over fitting to their preparation set. Irregular woodland is a kind of regulated AI calculation dependent on outfit learning. Gathering learning is a sort of realizing where you join various kinds of calculations or same calculation on different occasions to shape an all the more impressive forecast model. The irregular woodland calculation consolidates different calculation of a similar sort for example various choice trees, bringing about a timberland of trees, thus the name "Arbitrary Forest". The irregular woods calculation can be utilized for both relapse and order errands.

4. Proposed Methodology

It turned out to be certain that an individual model can't distinguish malicious site effectively and henceforth another methodology came into

presence. A Hybrid based model methodology is proposed to resolve the issues that emerges due to phishing sites. An Mixture based model is acquired by consolidating various models that improves the accuracy to recognize phishing assault. The beneath graph is a portrayal of the means in the proposed model. The dataset identified with phishing is gathered from the UCI archive. UCI archive is a gathering of information bases, area hypotheses that is openly accessible for investigation. credits are figured out from malicious sites. Dataset is arranged into preparing and testing dataset. Preparing and testing dataset are provided to a few classifiers like Logistic Regression, Decision Tree, SVM, Instance based figuring out how to assess their precision. Initially classifiers are dissected dependent on lone execution, at that point the classifiers with great outcomes i.e., better accuracy and less mistake rate are arranged. At that point we intertwine these best classifiers one by one to acquire the Hybrid characterization model.Add new heuristic features with machine learning algorithms to reduce the false positives in detecting new malicious sites. Made an attempt to identify the best machine learning algorithm to detect phishing sites with high accuracy than the existing techniques. Used three machine learning algorithms (Logistic regression (LR), support vector machine (SVM) and Decision Tree) to classify the websites as legitimate and Malicious. Based on the experimental observations, Decision tree outperformed the others. The choice of considering these machine learning algorithms is based on the classifiers used in the recent literature. Ensemble Methods: fuses various estimators' base predictions. Improves the robustness and generality of estimators. Many effective ensemble methods are available, among them these are the three representative methods

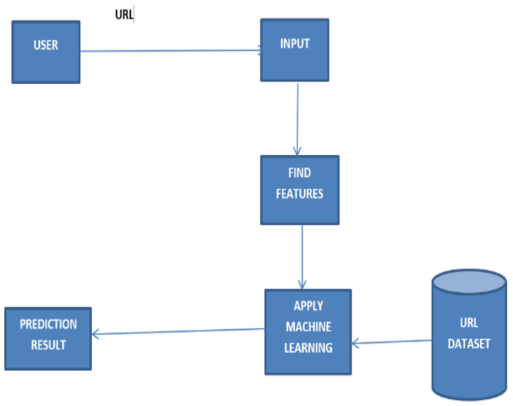
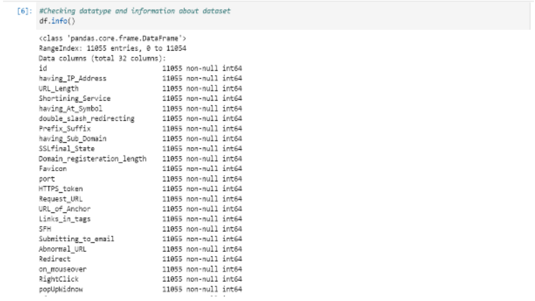


Figure 3: Architecture of Proposed model

4. Results

Module 1

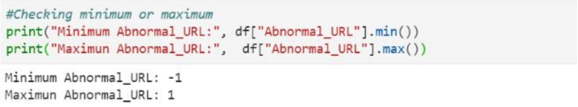
Data Validation Process and Preprocessing Technique



```
[6]: #Checking datatype and information about dataset
     df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 11055 entries, 0 to 11054
     Data columns (total 32 columns):
     id                            11055 non-null int64
     having_IP_Address             11055 non-null int64
     URL_Length                    11055 non-null int64
     Shortining_Service            11055 non-null int64
     having_At_Symbol              11055 non-null int64
     double_slash_redirecting      11055 non-null int64
     Prefix_Suffix                 11055 non-null int64
     having_Sub_Domain             11055 non-null int64
     SSLfinal_State                11055 non-null int64
     Domain_registeration_length   11055 non-null int64
     Favicon                       11055 non-null int64
     port                          11055 non-null int64
     HTTPS_token                   11055 non-null int64
     Request_URL                   11055 non-null int64
     URL_of_Anchor                 11055 non-null int64
     Links_in_tags                 11055 non-null int64
     SFH                           11055 non-null int64
     Submitting_to_email           11055 non-null int64
     Abnormal_URL                  11055 non-null int64
     Redirect                      11055 non-null int64
     on_mouseover                  11055 non-null int64
     RightClick                    11055 non-null int64
     popUpWidnow                   11055 non-null int64
```

Snapshot 4.1.1: Checking datatype and information about dataset



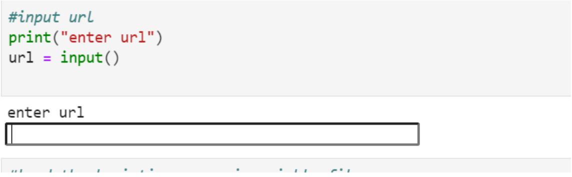
```
#Checking minimum or maximum
print("Minimum Abnormal_URL:", df["Abnormal_URL"].min())
print("Maximun Abnormal_URL:",  df["Abnormal_URL"].max())

Minimum Abnormal_URL: -1
Maximun Abnormal_URL: 1
```

Snapshot 4.1.2: Checking minimum or maximum Abnormal_URL

Module 2

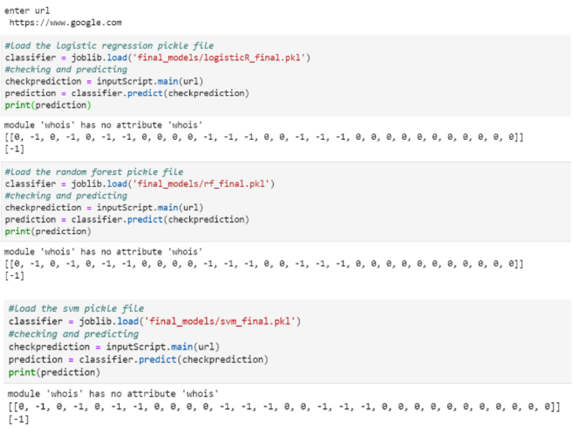
Detection of Malicious or Non-malicious URL



```
#input url
print("enter url")
url = input()

enter url
```

Snapshot 4.2.1: ULR Tab



```
enter url
 https://www.google.com

#Load the logistic regression pickle file
classifier = joblib.load('final_models/logisticR_final.pkl')
#checking and predicting
checkprediction = inputScript.main(url)
prediction = classifier.predict(checkprediction)
print(prediction)

module 'whois' has no attribute 'whois'
[[0, -1, 0, -1, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[-1]

#Load the random forest pickle file
classifier = joblib.load('final_models/rf_final.pkl')
#checking and predicting
checkprediction = inputScript.main(url)
prediction = classifier.predict(checkprediction)
print(prediction)

module 'whois' has no attribute 'whois'
[[0, -1, 0, -1, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[-1]

#Load the svm pickle file
classifier = joblib.load('final_models/svm_final.pkl')
#checking and predicting
checkprediction = inputScript.main(url)
prediction = classifier.predict(checkprediction)
print(prediction)

module 'whois' has no attribute 'whois'
[[0, -1, 0, -1, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[-1]
```

Snapshot 4.2.2: Different Algorithms detection result

Input: enter the URL

Output: Detection of malicious or non-malicious URL in different Algorithms.

Module 3

Exploration data analysis of pre-processing technique

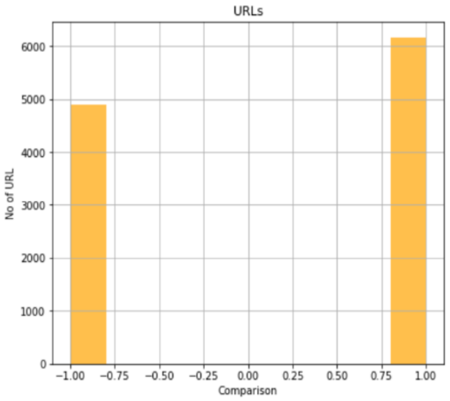


Figure 3.3 Comparison bar graph of Malicious v/s Non malicious

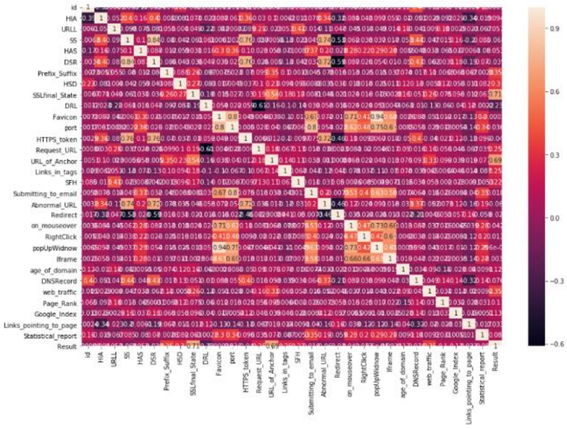
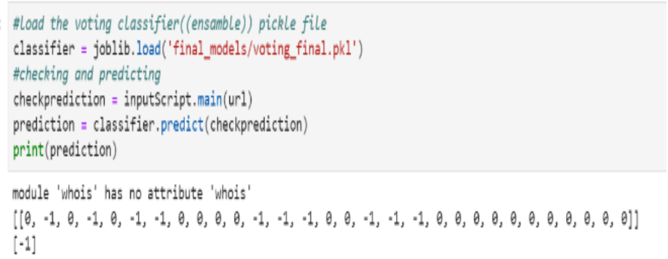


Figure 3.4: Heatmap of Various Parameters of Dataset

Module 4

Voting or Result of Ensemble technique



```
#load the voting classifier((ensamble)) pickle file
classifier = joblib.load('final_models/voting_final.pkl')
#checking and predicting
checkprediction = inputScript.main(url)
prediction = classifier.predict(checkprediction)
print(prediction)
```

```
module 'whois' has no attribute 'whois'
[[0, -1, 0, -1, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[-1]
```

Input: Uploading Voting_final.pkl file

Output: Result of ensemble Technique are malicious or Non-malicious URL

Conclusion and Future work

Malicious URL detection plays a critical role for many cyber security applications, and clearly machine learning approaches are a promising direction. Three different machine learning approaches (SVM, Logistic Regression, Random Forest)were taken to test the URL in which all Provided the same result according to the URL. This paper lets us the visualization of the dataset in the form of of graph as number of URL's and the malicious or Non-Malicious URL's comparison. Future scope of improvement would be removing unwanted popup information and fake URL's and also to optimize the work to implement in artificial Intelligence environment.

References

[1] Hossein Shirazi, Kyle Haefner, Indrakshi Ray Department of Computer Science Colorado State University Fort Collins, USA Email: {shirazi, kyle.haefner , iray}@colostate.edu.

[2] Srushti Patil Department of ComputerEngineering Sardar Patel Institute of Technology Mumbai, India srushti.patil@spit.ac.in, Sudhir Dhage Department of Computer Engineering Sardar Patel Institute of Technology Mumbai, India sudhir_dhage@spit.ac.in. [3] S. Gautam (✉) · K. Rani · B. Joshi Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, India e-mail: isudhanshugautam@gmail.com K. Rani e-mail: kritika.rani17@gmail.com B. Joshi e-mail: bansidhar.joshi@jiit.ac.in

[4] B B. B. Gupta gupta.brij@gmail.com 1 National Institute of Technology Kurukshetra, Kurukshetra, India.

[5] Chunlin Liu State Key Lab of Software Development Environment School of Computer Science and Engineering, Beihang University Beijing, China jackliu@buaa.edu.cn,Bo Lang State Key Lab of Software Development Environment School of Computer Science andEngineering, Beihang University Beijing, China langbo@buaa.edu.cn.

[6] D. Patil, and J. Patil, "Malicious URL detection using decision tree classifiers and majority voting technique," in Cybernetics and Information Technologies, vol.18, Issue.1, pp.11-29, March.2018. DOI:10.2478/cait-2018-0002

[7] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, "A new method for detection of phishing websites: URL detection," Proc. 2n d International Conference on Inventive Communication and Computational Technologies, Coimbatore, India, pp.949-952, 2018.

DOI:10.1109/ICICCT.2018.8473085

[8] Y. Huang, J. Qin, and W. Wen, "Phishing URL detection via capsule-based neural network," Proc. 13th International Conference on Anti-counterfeiting, Security, and Identification, Xiamen, China, pp.22-26, October, 2019. DOI:10.1109/ICASID.2019.8925000

[9] Y. Huang, Q. Yang, J. Qin, and W. Wen, “Phishing URL detection via CNN and attention-based hierarchical RNN,” Proc. 18t h International Conference on Trust, Security and Privacy in Computing and Communications and 13t hInternational Conference on Big Data Science and

Engineering, Rotorua, New Zealand, pp.112-119, August,2019.DOI:10.1109/TrustCom/BigDataSE.2019.0024

[10] S. Shivangi, P. Debnath, K. Sajeevan, and D. Annapurna,“Chrome extension for malicious URLs detection in social media applications using artificial neural networks and long short term memory networks,” Proc. 18t h International Conferences on Advances in Computing, Communications and Informatics, Bangalore, India, pp.1993-1997, September,2018DOI:10.1109/ICACCI.2018.8554647

[11] A. Vazhayil, R. Vinayakumar, and K. P. Soman, “Comparative study of the detection of malicious URLs using shallow and deep networks,” Proc. 9t h International Conference on Computing, Communication and Networking Technologies, Bangalore, India, pp.1-6, July, 2018. DOI:10.1109/ICCCNT.2018.8494159.

[12] Ignacio Arnaldo, Ankit Arun, Sumeeth Kyathanahalli, and Kalyan Veeramachaneni. 2018. Acquire, adapt, and anticipate: continuous learning to block malicious domains. In 2018 IEEE International Conference on Big Data (Big Data). IEEE, 1891–1898.

[13] A Astorino, A Chiarello, M Gaudioso, and A Piccolo. 2016. Malicious URL detection via spherical classification. Neural Computing and Applications (2016).

[14] Alejandro Correa Bahnsen, Ivan Torroledo, Luis David Camacho, and Sergio Villegas. 2018. DeepPhish: Simulating Malicious AI. In Proceedings of the Symposium on Electronic Crime Research, San Diego, CA, USA. 15–17.

[15] Sushma Nagesh Bannur, Lawrence K Saul, and Stefan Savage. 2011. 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.