Global Scientific JOURNALS

GSJ: Volume 10, Issue 9, September 2022, Online: ISSN 2320-9186
www.globalscientificjournal.com

DESIGN AND IMPLEMENTATION OF A MODEL FOR DETECTION OF MALICIOUS UNIFORM RESOURCE LOCATOR (URL) USING MACHINE LEARNING

[1]ANANTI, Henry Onyeka
henryananti@yahoo.com

[2] MGBEAFULIKE, Ike J
Ike.mgbeafulike@gmail.com

[1, 2] Department of Computer Science, ChukwuemekaOdumegwuOjukwu University, Uli

ABSTRACT

Malware is a worldwide scourge, and studies imply that with the advancement of the internet, it is getting even worse. Every minute a new virus is introduced, and while numerous practical ideas and tactics, such as the blacklist, have been suggested to lessen and eliminate cyber dangers, they have all been shown to be ineffective. In this study, we use statistical techniques to gather character distribution features and structural features of harmful URLs with a focus on character features. Then, we cross-train two different classifiers to find an efficient classifier for harmful URL detection. The study's objective was to develop and use a machine learning model for the detection of malicious URLs. The Python programming language and the Scikit-learn machine learning algorithm were used. Over 11054 datasets (both malicious and safe) were downloaded from Kaggle and stored in Excel to train the model. The Object-Oriented Analysis and Design Methodology (OOADM) was applied. In the end, the model worked well by maintaining a high level of accuracy and efficiency while displaying if a URL was malicious or benign. A parallel changeover was also recommended after deployment to avoid disrupting the existing process.

Keywords: Malware, Dataset, Internet, URL, Machine Learning

1.1 INTRODUCTION

Background to the Study

Hackers frequently employ phishing and spam to deceive users into clicking harmful URLs, infecting their systems with Trojans, or leaking the victims' private information(Lemay, 2020). Resources on the Internet are identified by their Uniform Resource Location (URL). It can be characterized into two basic components; The protocol identifier, which specifies the protocol to use, Resource name, which includes the IP address or domain name of the resource's physical

location(Sahoo*et al*,2017). Attackers typically make attempt to change one or more elements of the URL's structure in order to deceive visitors into visiting their harmful URL because each URL has a unique structure and format.Links that have adverse effects on the users are referred to as malicious URLs. These URLs may link users to undesirable, harmful, phishing, or pages where malware can be downloaded or where attackers can run code on users' systems. Even download links that are thought to be secure can contain malicious URLs. Through the transmission of files and messages on shared networks, these URLs can spread quickly. Attacks that use malicious URLs include drive-by downloads, phishing, social engineering, and spam (Heartfield, 2015). From statistical data from the 2019 according to Symantec's Internet Security Threat Report (ISTR), among the top 10 attack methods are attacks that propagate malicious URLs. The threat level and frequency of assaults utilizing the three primary URL spreading techniques—malicious URLs, botnet URLs, and phishing URLs—are increasing, according to this statistic.Additionally, it is evident from statistics that show a rise in the number of harmful URL distributed over the course of several years that approaches and methods to identify and stop these malicious URLs must be researched and put into practice.

1.2 Statement of the Problems

Sadly, 70% of websites contain flaws that hackers can use to get unauthorized access to sensitive back-end data, including financial data and consumer information.If organizations do not consistently focus on web security, they frequently become the target of hostile assaults.Many website owners overestimate the cost of safeguarding their servers and websites from hacking while underestimating the risk. Hence, it is crucial to realize that keeping reasonable and standard security is not expensive, but being hacked is.

a. Incessant increase in online fraud.

b. Statistical increment over the consecutive in the frequency of harmful URL distributions.

c. Use of Blacklist approach by most of the existing tools. The basis of this method for identifying malicious URLs is a set of indicators or guidelines that can promptly and accurately detect dangerous URLs.Nevertheless, this technique is unable to identify new dangerous URLs that do not match the specified signs or rules.

1.3 Aim and Objectives of the Study

The study's objective is to create and put into practice a machine learning model for detecting harmful Uniform Resource Locators (URLs). The system is designed with the following objectives:

a. To help Users identify malicious URL.
b. To increase or maintain maximum prediction accuracy of at least 97%.
c. To make sure a website visit is free of risks.
d. To develop a system that eradicates Blacklist approach applicable in most of the existing tools.
e. To develop a User friendly interface for non-professionals.

1.4 Review of Related Works

Over the year,over 90% of visitors depend on a website's appearance to determine its validity, attackers work to make malicious websites look almost exactly like trustworthy ones from a visual standpoint (Greenstadt, 2019). As a result, the researchers attempt to distinguish between trustworthy and fraudulent websites using the similarity of websites as a significant factor.
Previous studies have shown that some approaches employ website content similarity while others use visual similarity. Below are reviews of related words;

“Robust URL Classification with Generative Adversarial Networks” was implemented by Drago *et al,* in 2018. They employed datasets of log files gathered by Tstat and applied GAN for URL classification. The Real datasets acquired from log files, which were incredibly accurately identified for three safe datasets, were the research's main benefits. However, it is poorly classified for malware dataset.

“DeepPhish: Simulating Malicious AI” was implemented by Bahnsen*et al,* in 2018. By applying an LSTM network, threat actors were identified and the DeepPhish method was presented to show a probable attack. The advantages of their study effort boosted each threat actor's effectiveness rate; effectiveness rate is determined by the proportion of total URLs created using the same technique that eluded detection systems. However, the drawback is that there is not enough data to model success. The percentage of URLs that actually stole user data is used to gauge success.

Steve Sheng *et al.*2016 made use of the principles of learning science to design and iteratively improve an online game that instructs users on ethical practices to assist them prevent phishing attempts.The players' ability to discern between a phishing website and a genuine one was assessed both before and after playing the planned game, which involved playing the game itself

and reading ten articles about phishing.The outcomes demonstrate that playing the game can improve a participant's ability to identify a phishing website. NalinAsanka and colleagues (2016) developed a mobile version of a game that encouraged users of home computers to be vigilant about phishing threats in order to increase avoidance behavior Deanna et al 2018.

In order to determine the host-based and lexical characteristics of URLs from malicious web sites, Ma et al. 2019 outlined a technique based on statistical methods for categorizing URLs.To categorize malicious web sites on a larger scale, they employ lexical features of URLs as well as registration, hosting, and geographical data of the relevant hosts. By automatically extracting and evaluating tens of thousands of factors that could be indicative of suspicious URLs, these techniques are capable of creating highly predictive models.By using only their URLs, the generated classifiers are 95-99% accurate in identifying a significant number of malicious websites.However, their strategy necessitates a substantial feature set and relies on external servers to collect host information.

There are nine different types of machine learning techniques, such as ,AdaBoost, Random Forests, Neural Networks, Naive Bayes, and Bayesian Additive Regression Trees, Random Forests, Neural Networks, were covered in detail by Miyamoto et al. in 2018. On the cutting-edge CANTINA dataset, where they evaluated each classifier's performance, AdaBoost generated the most accurate results at 91.34%. They used a wide range of classifiers, but were unable to guarantee the durability of the solution due to the adaptive nature of these attacks and the lack of an updated dataset.The method Chen et al. (2018) suggested for determining the visual resemblancebetween two web pages.They put their technique to the test on the top visited websites to see how well it would work in practice. True positive and false positive accuracy rates were 100% and 80%, respectively.

Earth Mover's Distance (EMD) was utilized by Fu et al. in 2016 to assess how visually similar two webpages are. They used color and coordinate attributes to define the picture signatures after transforming the involved web pages into low-resolution photographs. Then, using EMD, the signature distances of the web page photographs were calculated.In order to determine whether a web page was phishing or not, they used an EMD threshold vector.Additionally, they developed a real system that is currently in use online and has successfully stopped several actual phishing incidents.

1.5 Summary of Literature Review and Knowledge Gap

Many cyber security applications rely heavily on malicious URL identification, and it is obvious that machine learning methods represent a promising future. In the course of this study, a thorough and methodical assessment on malicious URL identification was carried out utilizing machine learning approaches. Then, we talked about taking into account prior work for harmful URL identification, mainly in the form of creating new feature representations and learning algorithms for addressing the harmful URL detection problems. We provided a specific methodical formulation of malicious URL detection from a machine learning perspective.

In this research, we mostly categorized, if not all, the previously published works on harmful URL detection, and as well identified the needs and difficulties in creating a service for malicious URL detection in actual cyber security applications. Lastly, we emphasized certain practical concerns for the application domain and showed some significant outstanding issues that require additional research study.In particular, automated identification of dangerous URLs using machine learning remains a highly difficult open topic, despite the vast studies and the impressive progress made in the previous few years.

Future suggestions include improved feature extraction and representation learning, effective machine learning algorithms for teaching predictive models, especially for trying to tackle concept drifts (e.g., effective online learning) and other emerging issues (e.g., domain adaptation when attempting to apply a model to a new domain), and finally a clever design of closed-loop system of obtaining labeled data and user responses (e.g., incorporating an online active learning approach in a real system)

1.6 Analysis of the New System

In order to solve these issues, researchers have been making use machine learning methods for malicious URL identification for the past ten years. Machine learning approaches provide a prediction function to categorize a URL as harmful or safe based on the statistical attributes using a set of URLs as training data. This makes it possible for them to generalize to new URLs, unlike blacklisting techniques. The key need for training a machine learning model is the availability of training data. This would be relevant to a collection of various URLs in the context of the detection of dangerous URLs. In general, there are three categories of machine

learning: supervised, semi-supervised, unsupervised, which, respectively, mean that the training data has labels, does not have labels, and only has labels for a small fraction of the training data.Labels allow users to determine if a URL is harmful or benign. The next stage is to extract significant features from the training data that machine learning models can utilize to analyze mathematically while also adequately describing the URL.For instance, learning a solid prediction model from the URL string alone could be difficult (which in some extreme conditions may lessen the prediction model to a blacklist method).

Therefore, in order to create a good feature representation of the URL, and on the basis of some rules or heuristics, one would have to select the appropriate features. Host-based features (WHOIS data, geo-location characteristics, etc.), lexical features (statistical aspects of the URL string, a bag of words, an n-gram, etc.), and other elements may be used. These features, along with others used for this work, will be described in great detail in this survey. These features must b (such as a numerical vector) so that they may be used to train a model using off-the-shelf machine learning method. Since a fundamental premise of machine learning (classification) models is that harmful and benign URL feature representations have different distributions, it is imperative that these features have the ability to provide pertinent information. The quality of the final machine learning-based malicious URL predictive model depends greatly on how well the URLs' features are represented. The next step in developing a prediction model utilizing training data and the appropriate feature representation is to actually train the model. Numerous classification methods are available that can be applied directly to training data (Naive Bayes, Support Vector Machine, Logistic Regression, etc.).

But several features of the URL data could make the training difficult (both in terms of scalability and learning the right concept). For instance, there could be millions of URLs available for training (or even billions). Therefore, the training period for conventional models can be too long to be useful. Consequently, for this task, a family of scalable learning techniques called online learning has been actively utilized. The limited representation of Bag-of-Words (BoW) features in the URLs presents another challenge. Every word that could possibly appear in any URL becomes a feature because these features show whether a specific word (or string) appears in a URL or not. There is a chance that this method will result in millions of sparse features (most features are absent most of the time because a URL typically only contains a small fraction of the millions of words that may be used). In order to increase learning effectiveness

and efficiency, a teaching approach should take use of this sparsity. There are additional difficulties that are unique to this task, and they have called for relevant advancements in machine learning methods to lessen these difficulties.

In this study, we examine cutting-edge machine learning methods for identifying malicious URLs. We pay particular attention to the developments of learning algorithms and feature representation in this field. We systematically classify the many feature representations that were utilized to generate the training data for this job, as well as the various learning techniques that were employed to develop an effective prediction model. We also address unsolved research issues and suggest future research possibilities. Blacklists (including heuristics) and Machine Learning are the two primary kinds of tactics used to identify dangerous URLs. We define the circumstance as a machine learning problem where the main need is for proper feature representation and the learning method is used. We then show various feature representation types that have been applied to this problem in detail. Then, several algorithms that have been built based on the characteristics of URL data and have been used to solve this problem are shown.

1.7 Methodology Adopted

Analysis simply refers to a detailed, thorough examination of the structure of something, elements, system requirements. Hence, System Analysis and Design involves ascertaining the objectives and problems of the existing system, and proper analysis carried out on facts gathered. System analysis can be simplified here as a detailed inquiry carried out by the system analyst to identify a better course of action and make a better decision on the proposed system. The methodology adopted for the research is the Object Oriented Analysis and Design Methodology (OOADM).This method makes it possible to implement a software solution based on the ideas of objects.Object-oriented analysis, object-oriented design, and object-oriented implementation are the three main stages of software development utilizing the object-oriented methodology. Each phase focuses on a particular design concerns. The next major development in software engineering has been recognized as Object Oriented Development (OOD). Some of the advantages of object oriented methodology are;

a. It assures to shorten the duration of development.

b. Lessen the time and resources needed for maintaining current applications.

c. Reuse code more often.

d. provide organizations using it with a competitive edge.

1.8 System Design

The process of defining a system's architecture, product design, modules, interfaces, and data in order to meet predetermined requirements is known as systems design.You could view of systems design as the application of systems theory to the creation of products. This is how the suggested system is really designed.In order to translate the system, need into a representation of the system designed, many methodologies and ideas are applied in this chapter's system design.
The new system's goals are to assist users in identifying bad urls, shield people from online attacks, and make sure your website is safe and secure from risks.

1.9 Objectives of the Design

The objective of the proposed is to put the design carried out in the analysis into a functional system. The system will be able to perform the following functions:

i. to assist users in recognizing fraudulent URL.
ii. to guard against users being attacked by a malicious URL.
iii. To make sure your website is safe, secure and free of insecurities.

2.0 System Specifications

2.1.1 Math Specification

Evaluation Metrics:

i. **Accuracy:**the Percentage of correct decisions among all testing samples

$$\mathbf{ACC} = \frac{\mathbf{TP + TN}}{\mathbf{TP + TN + FP + FN}} \mathbf{X\ 100\%}$$

Where:

TP = True Positive is the number of malicious URLs correctly labeled

FN = False Negetive is the number of malicious URLs misclassified as Safe

TN = True Negative is the number of safe URL correctly labeled.

FP = False Positive is the of safe URLs misclassified as malicious

ii. **Precision:** is the percentage of malicious URLs correctly labeled (TP) among all malicious URLs labeled by the classifier**(TP + FP)**

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%$$

iii. **Recall:** is the percentage of malicious URLs correctly labeled (TP) among all malicious URLs of the testing data (TP + FN)

$$\text{Recall} = \frac{TP}{TP + FP} \times 100\%$$

2.1 Input/Output Format

Input Format: Input design determines greatly the nature of the desired output. The desired input will yield the output being expected. The input design has to do with the structure, nature and format of input that the system needs for its proper functionality. Here, the User is expected to key in the URL he or she wishes to verify the authenticity in the taskbar and then click on the "Check" button to verify. The system then processes the data provided before displaying the output.

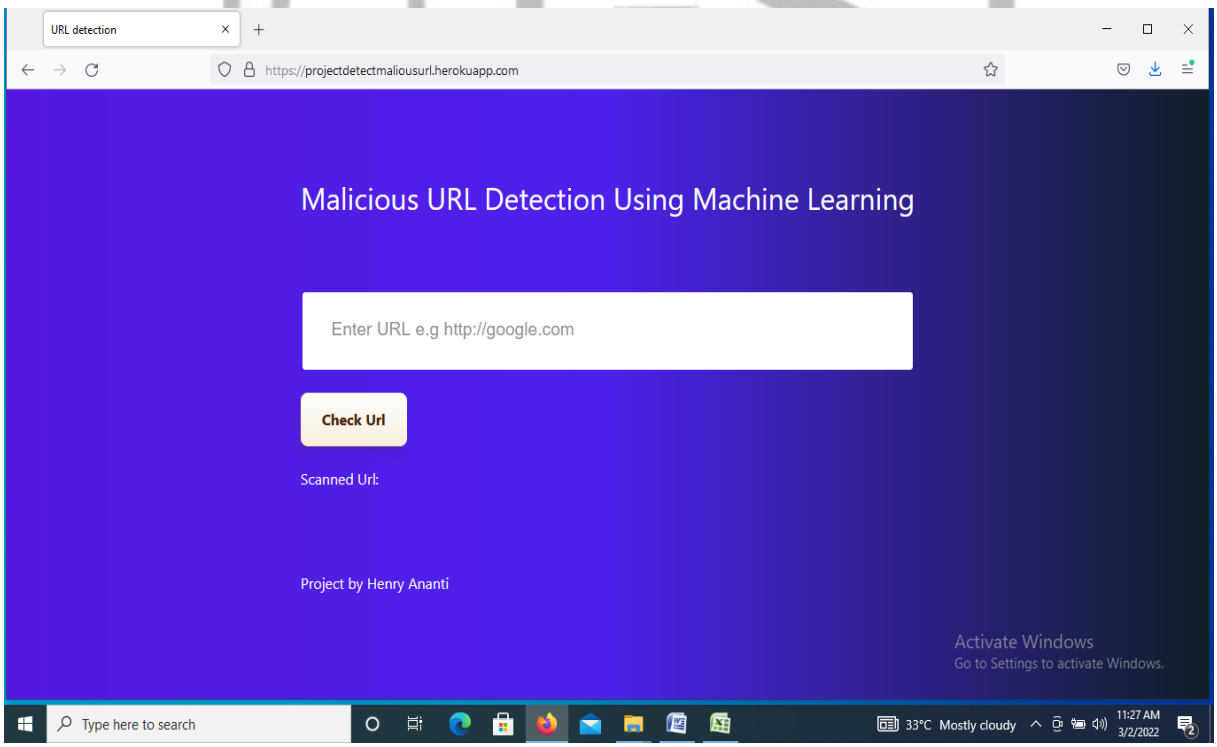

Fig 4.1 Input Display of the new System

Output Format:

A system is determined by how the reliability and acceptability of its output quality as a major factor. Its interface is user friendly. It saves the user the efforts of remembering too many details at a time. The user interface has a section where the user types the URL for confirmation, a Check button for to execute the verification and then a Caution button that notifies the user if he or she wishes to continue.

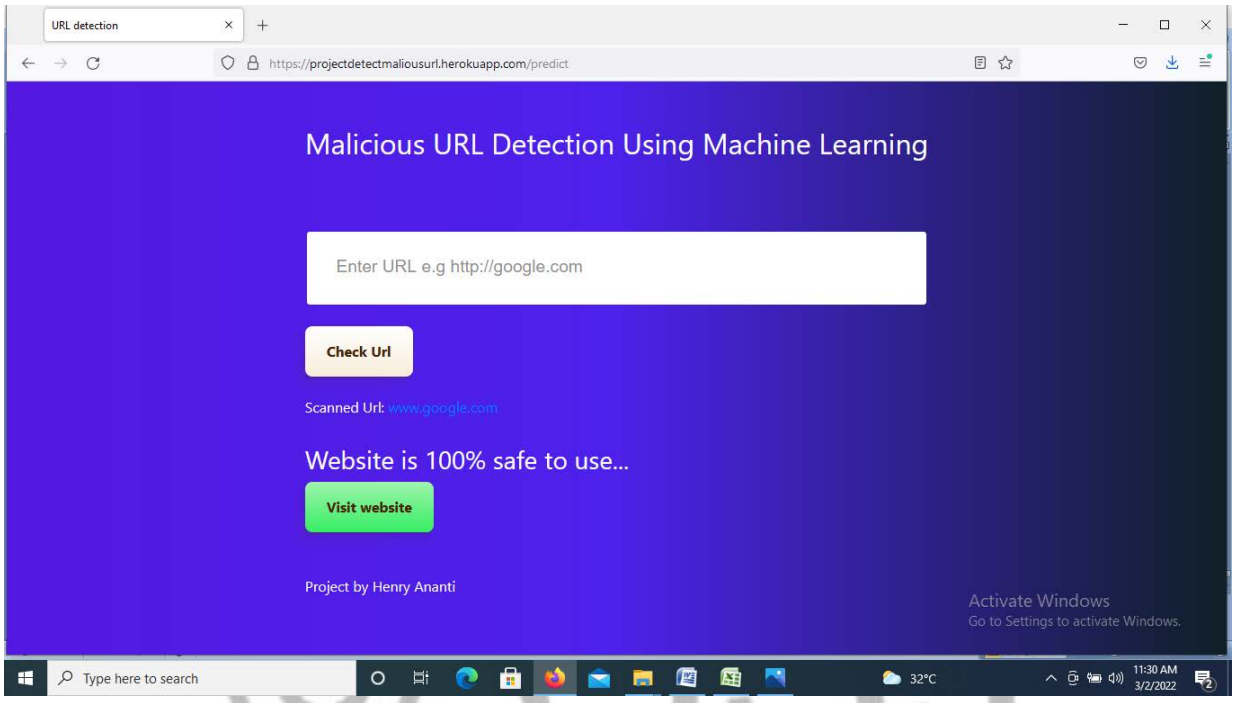

Fig 4.2 Output Display of the new System

2.2 Summary

This research was aimed at providing an efficient and reliable model that will assist users and organization at large in detecting malicious URls in order to avert attack from cybercriminal from gaining illicit access to her data. The research was commenced with background information of the study and limitations were analyzed. With the introduction of machine learning, the new system is faster and more reliable compared to blacklist approved.

2.3 Conclusion

In this work, we investigated the issue of detecting phishing using machine learning methodologies and designed and implemented a model for the identification of harmful URL using machine learning. In our initial attempt, we looked at the issue from a statistical point of view and discovered a number of intriguing patterns. We learned from the experience that solving a problem solely from a statistical standpoint is insufficient. For a solution to be

successful, the attacker's motivation must also be taken into account.We discussed the initial strategy for developing features that are solely dependent on domain names in order to identify harmful websites using machine learning.The emphasis of feature design was on the avoidance of any classification bias resulting from differently selected datasets of legitimate and phishing pages.

Our method is unique compared to all earlier efforts in this field since it examines the connection between the domain name and the phishing intention. With cross-validated data, we are able to attain a classification rate of 97% using just seven characteristics. In addition, we were able to demonstrate a 97-99:7% detection rate for live blacklisted URLs from kaggle.com. This demonstrates how flexible our approach is to the sophisticated tactics hackers employ to circumvent such detection systems. An enemy may create a page in an effort to get around our strategy that will make users question its motives.

Additionally, we showed the drawback of using URL attributes like URL lengths, which appear to provide higher accuracy but might not in the near future. Our quick feature extraction and classification speeds demonstrate the suitability of our method for implementation in real time.
Our strategy is probably going to be quite effective in contemporary phishing techniques like extreme phishing that are meant to trick even knowledgeable users.

References

Afroz S.&GreenstadtR.(2019). PhishZoo: Detecting PhishingWebsites by Looking at Them. In 2019 IEEE Fifth International Conference on Semantic Computing, pages 368–375, September 2019.

AnandA., GordeK., MonizJ. R. A., ParkN., ChakrabotyT., &ChuB. (2018), “Phishing URL detection with oversampling based on text generative adversarial networks,” Proc. International Conference on Big Data, Seattle, USA, pp.1168-1177, December, 2018. DOI:10.1109/BigData.2018.8622547

André Bergholz, Jan De B., Sebastian Glahn, Marie-Francine Moens, Paaß Gerhard, &SiehyunStrobel. New filtering approaches for phishing email. Journal of Computer Security, 18(1):7–35, 2010.57

Ankit Kumar Jain &XGuptaB. B.(2016). A novel approach to protect against phishing attacks at client-side using auto-updated white-list. EURASIP Journal on Information Security, 2016(1):9, 2016.

Ankit Kumar Jain and GuptaB. B. (2017) .Towards detection of phishing websites on client-sideusing machine learning based approach. Telecommunication Systems, Dec 2017.

APWG.(2017) Global phishing survey: Trends and domain name use in 2016, 2017. [Online; accessed 21-October-2016.

ArnaldoI., ArunA., &KyathanahalliS.(2018), "Acquire, adapt and anticipate: continuous learning to block malicious domains," Proc. International Conference on Big Data, Seattle, USA, December, 2018. DOI:10.1109/BigData.2018.8622197

Choon Lin Tan, Kang LengChiew, KokSheik Wong, &San Nah Sze (2016).Phishwho: Phishing webpage detection via identity keywords extraction and target domain name finder. DecisionSupport Systems, 88(C):18–27, 2016.

Corbett & Philip B. (2016). "It's Official: The 'Internet' Is Over". *The New York Times.*ISSN 0362-4331.Archived *from the original on 14 October 2020*.Retrieved 29 August 2020.

CovaM., KruegelC., and VignaG. (2018), "Detection and analysis of drive by download attacks and malicious javascript code," in Proceedings of the 19th international conference on World wide web. ACM, 2018, pp. 281–290.

Daisuke Miyamoto, Hiroaki Hazeyama, &YoukiKadobayashi (2018).An evaluation of machine learning-based methods for detection of phishing sites. In International Conference on Neural Information Processing, pages 539–546.Springer, 2018.

David Bremner, Erik Demaine, Jeff Erickson, John Iacono, Stefan Langerman, Pat Morin, &Godfried Toussaint (2015). Output-sensitive algorithms for computing nearest-neighbour decision boundaries. Discrete & Computational Geometry, 33(4):593–604, 2015.63

Sahoo .D, Liu C. &HoiS.C.H. (2017), "Malicious URL Detection using Machine Learning: A Survey". CoRR, abs/1701.07179, 2017.

Samuel Marchal, Giovanni Armano, TommiGrondahl, KalleSaari, Nidhi Singh, &AsokanN.(2017). Off-the-hook: An efficient and usable client-side phishing prevention application. IEEE Trans. on Computers, 66(10):1717–1733, 2017.

Samuel Marchal, KalleSaari, Nidhi Singh, &N Asokan (2016). Know your phish: Novel techniquesfor detecting phishing sites and their targets. In Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on, pages 323–333. IEEE, 2016.58

ShiY., ChenG.,&LiJ.(2018), "Malicious domain name detection based on extreme machine learning," in Neural Processing Letters, vol.48, pp.1347-1357, 2018. DOI:10.1007/s11063-017-9666-7

Steve Sheng, Bryant Magnien, PonnurangamKumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, & Elizabeth Nunge (2018). Anti-Phishing Phil: The Design and Evaluation of a Game That Teaches People Not to Fall for Phish. In Proceedings of the 3rd Symposium onUsable Privacy and Security, SOUPS '07, pages 88–99, New York, NY, USA, 2018. ACM.