IP Reputation Analysis of Public Databases and Machine Learning Techniques

Jared Lee Lewis, Geanina F. Tambaliuc, Husnu S. Narman, and Wook-Sung Yoo
{lewis504}{tambaliuc}{narman}{yoow}@marshall.edu
Weisberg Division of Computer Science, Marshall University, Huntington, WV 25755

***Abstract*—As Internet usage is increasing worldwide, today's network is challenged with numerous cyber-attacks. An effective way to prevent users from cyber-attacks is to identify and create blacklists of those malicious domains. However, there are several issues related to the blacklist approach. Some authorized domains can mistakenly be added to blacklists, and some blacklist databases are not regularly maintained or updated. To solve these issues, we developed the Automated IP Reputation Analyzer Tool (AIPRA), a partly cross-checking system which automatically analyzes many reliable blacklist databases and assigns a weighted security degree of domains and IP addresses to inform users and applications about possibilities of malicious activities. However, there are some notable problems with blacklists, including false positives, inability to account for new malicious domains, and the constantly changing IP addresses of the malicious sites. To remedy this, we have tested four different machine learning approaches with several parameters, such as geolocation to analyze the performance of the approaches. Then, we integrate the geolocation-based machine learning approach into AIPRA to identify a malicious IP address or FQDN (Fully Qualified Domain Name). The results show that various public blacklist databases and machine learning techniques have significantly different results for the same set of IPs. While the results of machine learning methods can differ up to 25%, the blacklists result differ up to 80% differences for the same set of IPs. Therefore, our developed tool AIPRA is not only beneficial with crosscheck but also using machine learning to identify and eliminate the security issues which are caused by new harmful sites and outdated blacklists.**

***Index Terms*—Security; IP reputation; machine learning; blacklists**

I. Introduction

As Internet usage is increasing worldwide and many parts of our lives rely on the Internet, today's network is challenged with numerous cyber-attack, which consumes up to 80% of the data traffic with spam emails [1]. According to McAfee Lab report, five new malware samples are discovered per second in Q1 2018, which means more than one hundred and fifty million new malware samples are discovered per year. The attacks become huge problems for individuals, business, organizations, universities, and governmental agencies with economic loss and psychological damages. An unidentified phishing email or an unconscious click can cause unrecoverable damage to an organization [2], [3].

Several filtering techniques have been developed by different organizations to prevent all entities from such cyber-attacks [2], [4], and the Domain Name System (DNS) plays a vital role in filtering and protection techniques. The botnet used by threat actors, as an example, depends on DNS to infect and distribute malware to other users. An effective way to protect users from such threats is to identify and create blacklists of those malicious domains [5], [6]. Many private, commercial and open blacklist databases have been created [7]–[13]. However, there are several issues related to blacklists approach: some authorized domains can mistakenly be added to blacklists, and some blacklist databases are not regularly maintained or updated [14], [15]. Therefore, to solve these issues, Automated IP Reputation Analyser (AIPRA) [16] is developed to reduce the amount of the frauds, identity thief with phishing, and other related security problems. AIPRA is a partly cross-checking system which automatically analyzes several reliable blacklist databases and assigns a weighted security degree of domains and IP addresses to inform users and applications about possibilities of malicious activities. However, there are some notable problems with blacklists, including false positives [17], inability to account for new malicious domains, and the constantly changing IP addresses of the existing malicious sites. To remedy those problems, we have adopted four different machine learning approaches with and without geolocation parameters to analyze the effectiveness of the machine learning techniques. Then, we integrate the machine learning approach into AIPRA to identify a malicious IP address and Fully Qualified Domain Name (FQDN).

There are several proposed works on checking IP reputations with machine learning approaches [18]–[24]. In [19], a machine learning model relies on a deep neural architecture and is trained on a large passive Domain Name System (DNS) databases is presented. The model can detect 95 % of the malicious hosts with a false positive rate of 1:1000. However, the training time is significantly high due to large training data, and the delay information is not analyzed. In [18], a scalable and effective graph inference system based on the loopy belief propagation algorithm is introduced to detect malicious domains and IP addresses. The system detection rate is 86% and 87% domain and IP reputations, respectively. In [24], the performance between Local Outlier Factor (LOF) and Isolation Forest (iForest) is evaluated by probing the sim-

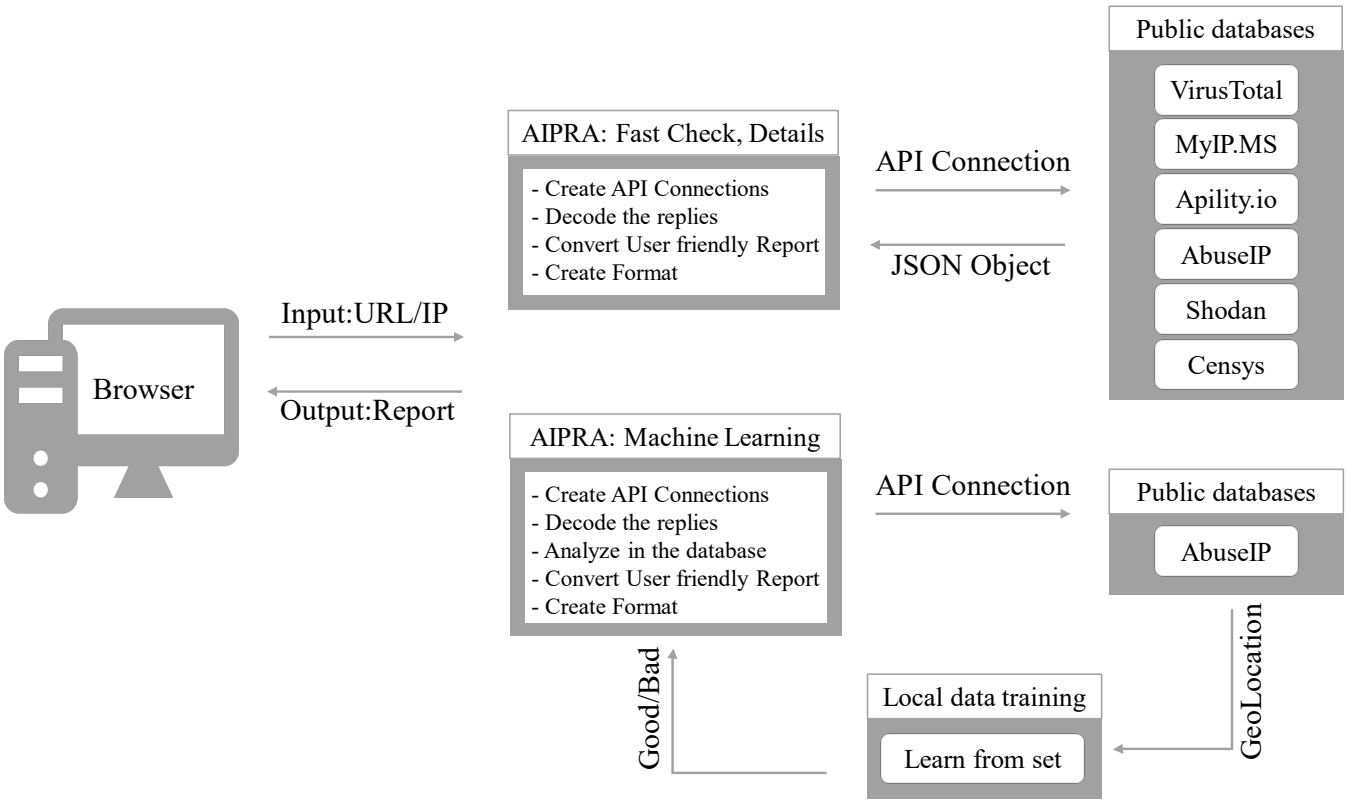

Fig. 1: System model for the AIRPA

ilarities and differences between the result of each approach. They found out that that iForest performs well in identifying anomalies compared to LOF. In [23], a novel learning evasive botnet architecture; and a stealthy and secure mechanism are introduced, and it is shown that it is difficult for a horizontal correlation learning algorithm to separate malicious email traffic from normal email traffic based on the volume features and time-related features with enough confidence. In [20], "Segugio" is introduced. It can track the occurrence of new malware-control domains with up to 85% true positives (TPs) at less than 0.1% false positives (FPs). However, true positives and false positives are based on only a set of 53 new domains which is a very small set to justify the correctness. In [22], a novel granular support vector machine - boundary alignment algorithm (GSVM-BA) is designed. GSVM-BA repetitively removes positive support vectors from the training dataset to look for the optimal decision boundary. There are other techniques such as mathematical based IP reputation [25], prefix technique to understand origins of IPs [26], efficient look-up techniques [27]. For further details on IP reputation, the short survey paper [21] can be read. It explains a number of different techniques besides machine learning techniques. The limitations of the above works are: machine learning training takes longer time due to large training sets, and blacklists can be outdated. Therefore, AIRPA eliminates limitations by using both approaches.

The *objective* of this paper is to analyze the public databases and machine learning techniques to detect malicious IP addresses and domains, and introduce AIRPA, which uses both approaches to check the reputations of IP and domains. The key *contributions* of this paper can be listed as follows:

- Automated IP Reputation Analyzer Tool (AIPRA) [16] is developed. It is a partly cross-checking system with integrated geolocation-based machine learning approach to automatically analyzes a number of reliable blacklist databases and assigns a weighted security degree of domains and IP addresses to inform users and applications about possibilities of malicious activities.
- Four public databases which are VirusTotal [8], MyIP.MS [11], AbuseIPDB [9], and Apility.io [10] are analyzed based on false-positive results for the same set of IPs and domains.
- Three machine learning algorithms which are Naive Bayes [28], Random Forest [29], and Logistic Regression [30] are analyzed with and without geolocation in terms of reputation.

The *results* show that a cross-checking system which automatically analyzes several reliable blacklist databases with a machine learning technique is the best approach to protect the online users. By this way, not only the problems with the

databases which are not regularly maintained or updated will be avoided, but also new malware websites can be detected.

The rest of the paper is organized as follows: In Section II, the system model is explained. In Section III, the experiment with analysis and results are presented, and finally, Section IV has the concluding remarks with future works.

II. System Model and Design

AIPRA is a web application that connects to several online databases through an API connection. The application checks URLs, domains, IP addresses, and provides a detailed report to users. The system consists of three major components: Web Interface, Connection to Public Databases, Machine Learning, and Analyzer, as shown in Fig. 1. The following subsections explain each component.

A. User Interface

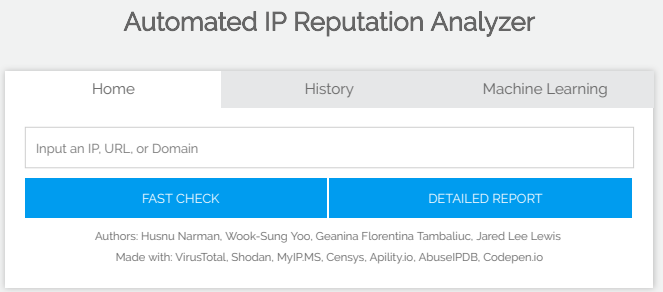

Fig. 2: User interface for the AIRPA

Fig. 2 shows the web application which users interact with the browser. The application has a simple interface which requests input from users (IP, URL or Domain), check it against multiple databases and retrieve information from them. The application provides detail reports from each database separately or the single result with a weighted score without detail information. If users cannot find the result in the databases, the machine learning feature can be used to obtain the prediction. We intentionally separate the public database result from the machine learning part to show the effectiveness of both blacklist and the used machine learning techniques. History in the user interface is used to check the three known IPs for our future works.

B. Public Databases

The web application is connected to six main databases selected from a list of twenty well-known databases, and 67 sub-databases, which are publicly available engines to check the reputation of IPs, URLs, and domains. The main six databases are VirusTotal [8], MyIP.MS [11], Censys [13], AbuseIPDB [9], Apility.io [10], and Shodan [12].

The web application retrieves JSON objects from the public databases containing the following information based on input (IP address, Domain name, or URL) entered:

- **VirusTotal [8]:** URL input: a security score for the URL entered by the user, ratings from each sub-database.
  IP input: ASN, owner, resolutions, detected/undetected URLs, detected/undetected downloaded samples, detected/undetected communicating samples.
- **MyIP.MS [11]:** Domain input: website, IP, owner, owner's address, phone number, cidr, host, popularity rank, sites, servers, IP change history.
  IP input: owner information. servers related to the IP, websites on the IP, total DNS, and OS on IP, total browsers and user agents on IP, popularity rank, number of visitors per day.
- **Censys [13]:** IP input: ports, tags, protocols, Regional Internet Registry (RIR), routed prefix, and other autonomous system details.
- **AbuseIPDB [9]:** IP input: IP networks, IP type (blacklisted or not), geolocation, categories of fraud related to the IP and when they were reported.
- **Apility.io [10]:** IP input: IP type (blacklisted or not) information.
  Domain input: general score and interpretation of it, domain databases checked.
- **Shodan [12]:** IP input: location, port, and hostname details.

Although the databases can both provide free and non-free services, the free services are limited. For example, currently, VirusTotal allows four requests per minute, AbuseIPDB allows 10000 requests per month, Apility allows 250 requests per day, and MyIP allows 150 requests per month. In AIPRA, the free versions have been used.

C. Machine Learning

A data set is required to train machine learning algorithms to identify malicious domains and IP addresses. There are several public databases such as AbuseIPDB provide a list of malicious IP addresses and FQDN [9]. In this research, Java Selenium Automation has been used to craw these sites and append to two separate SQLite databases. There are also resources for gathering non-malicious IP Addresses and FQDN's [31] in addition to the public databases such as search-engine blacklists. However, larger data sets with deep learning can result in a long time to train the algorithm while simpler algorithm can produce similar results [32]. Moreover, determining the maliciousness of the websites may require retraining. Two data sets, one for 80,000 FQDN's and one for 80,000 IP Addresses are collected to test the effectiveness in a reasonable time training. Because of the nature of the data, a binary classification approach is taken to label all entries as either "good" or "bad" where good means there is no malicious activity and bad means there is malicious activity. Both data sets are balanced with 40,000 entries labeled "bad" and 40,000 entries labeled "good". Geolocation information is also used to gather more information aside from just an IP address or FQDN. AbuseIPDB's free API is used to obtain

geolocation. Fig. 1 shows the system model of how this data set is used in the application.

1) Extracting Features: In both of the data sets, each FQDN or IP Address has city, zip code, region code, IP type (if an IP address), country code, and other related information. Likely, a particular IP address or FQDN does not have any geolocation information attached; in this case, this information is null. For both an FQDN and IP address, Term Frequency, Inverse Document Frequency (TFIDF) vectorization is used to split the entry into tokens. For example, an FQDN such as youtube.com/watch?v=8o5smgnl8wA is split into tokens: youtube.com, youtube, watch?v=8o5smgnk8wA. This type of vectorization applies a number to each token for how frequently it occurs in the data set, allowing specific features to mean more to the algorithm than others. In the remaining information discussed above, a one-hot encoding technique is used to apply a binary number to each.

2) Logistic Regression: A logistic regression algorithm is used to learn the data set and make predictions on new inputs. Logistic regression is implemented by using scikit-learn [30]. Logistic regression is an effective way to learn IP addresses and FQDN's as not only it is applicable to the data set, but it also effectively analyzes the relationship between all variables with respect to the binary dependent variable (good or bad). The following formula is used in a logistic regression implementation:

$$y = \frac{e^{b_0+b_1*x}}{1+e^{b_0+b_1*x}} \tag{1}$$

Where x is the input value (from 0 to 1), b_0 is the intercept value, b_1 is the coefficient, and y is the output value. The output value will be a prediction variable from 0 to 1, where any value above 0.5 means the IP address or FQDN is malicious.

D. Analyzer

The Analyzer is the central control part of the system. PHP is used to connect the Public Databases to the application by creating API connections. Then, the analyzer extracts data as JSON objects to be analyzed. Afterward, it performs statistical analysis according to the selected category and displays data to the user as well as store statistical information in the databases. Currently, the application has four main functionalities: Fast Check, Detailed Report, History for private IPs, and Machine Learning. Firstly, the Fast Check provides the user an easy and fast way to check an URL, IP, or Domain whether the entered address is blacklisted. Secondly, Detailed Report with an URL input provides both a security score and ratings from each sub-database. Furthermore, Detailed Report with an IP input provides users with a detailed report with the following information: ASN, owner, resolutions, detected/undetected URLs, detected/undetected downloaded samples, detected/undetected communicating samples, IP type (blacklisted or not), geolocation details, categories of fraud related to the IP, and when they were reported, port, hash, organization, internet service provider, protocols, cidr, servers, websites, and popularity rank. Also, Detailed Report with a Domain input provides users a report with the following information: website, IP, owner, owner's address, phone number, cidr, host, popularity rank, sites, servers, IP change history, security score, and databases in which the domain is blacklisted. Thirdly, the History for private IPs function provides users a monthly status history about specific IPs. Finally, machine learning algorithms are used to detect new malicious domains.

III. Analysis and Experiment

In this section, we explain the conditions of the experiment and analysis with the obtained results.

A. Limitations

The current application uses free versions APIs, and unfortunately, they have some limitations. VirusTotal [8] allows checking four IPs per minute, AbuseIPDB [9] allows checking 10000 IPs per month, Apility.IO [10] allows checking 250 IPs per day, and MyIP.MS [11] allows checking 150 IPs per month. In order to understand the performance of the public databases, we used the same set of IPs. However, the tested set sizes are different because of the limitation of the number of IPs in each database. Therefore, we normalized the obtained results based on the number of tested IPs in each public database.

B. Feature of Testing Server

A local server is created to test the efficiency of the machine learning techniques. The local server system information is Intel(R) Core(TM) i7-6700 CPU 3.40 GHz, 3.41 GHz with 16 GB RAM. The system uses 64-bit Windows 10 (version - 1903).

C. Public Database Comparison

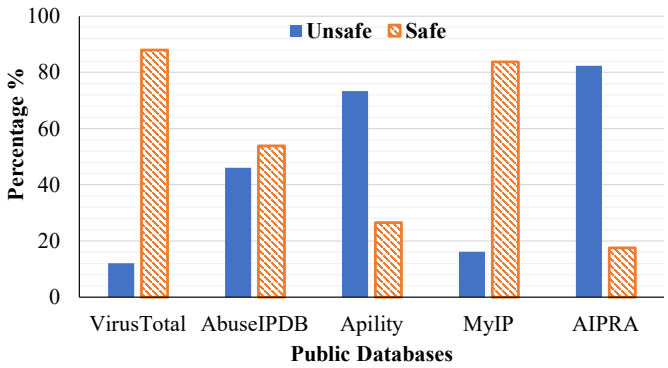

Fig. 3: Safe and Unsafe IPs detection from public databases.

A set of random 1586 IPs which are not safe are used to compare the efficiency of the public databases. Fig. 3 shows the obtained results from each database. VirusTotal

recognizes only 13% out of the tested IPs are unsafe and 87% as safe. MyIP also shows similar results with VirusTotal and detects 16% as unsafe while 84% as safe of IPs. On the other hand, AbuseIPDB has higher detection rate comparing to VirusTotal and MyIP. It detects 46% of IPs as unsafe and 54% IPs as safe. Apility has the highest detection rate, with 73% as unsafe and 27% as safe. On the other hand, AIPRA by using same databases (VirusTotal, MyIP, AbuseIPDB, and Apility) without the other databases mentioned previously (Censys [13] and Shodan [12]) can detect 82% of IPs as unsafe and 18% as safe. Therefore, the highest detection rate to slowest detection rate is AIPRA > Apility > AbuseIPDB > MyIP > VirusTotal. However, false positives can also be high in cross-checking if there is no elimination. In AIPRA, the detection is given based on the score, not just the detected or undetected.

D. Comparison of Machine Learning Methods

Three machine learning techniques which are Naive Bayes - multivariate Bernoulli models (NB) [28], Random Forest - with 100 estimators (RF) [29], and Logistic Regression (LR) [30] are analyzed with and without geolocation (Naive Bayes with geolocation (NBG), Random Forest with geolocation (RFG), and Logistic Regression with geolocation (LRG)) in terms of correct detection and running time in the local server. 2000 IPs (1000 good and 1000 bad) were tested with and without geolocation by using the mentioned machine learning techniques after training with 80,000 + 80,000 as explained in Section II-C.

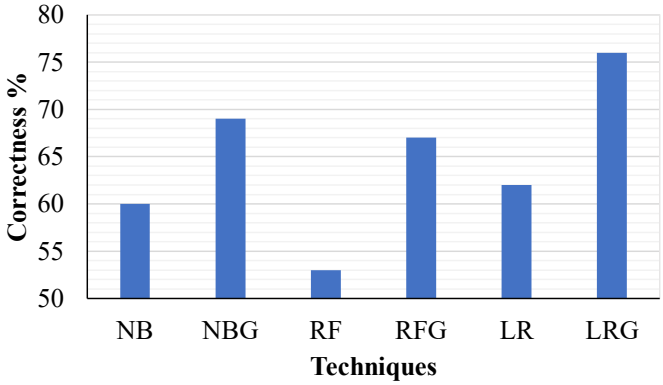

Fig. 4: Correct IP detection rate for machine learning techniques.

1) Correct Detection Rate: Fig. 4 shows the correct detection rate of the three techniques with and without geolocation. When only considering the features extracted from an IP address, 60% accuracy is achieved in NB, 53% accuracy is achieved in RF, and 62% accuracy is achieved in LR. With geolocation information, 69% accuracy is achieved in NB (NBG), 67% accuracy is achieved in RF (RFG), and 76% accuracy is achieved in LR (LRG). While the lowest effect of geolocation is in NB with 9%, the highest effect of geolocation is in RF and LR with 14%. Therefore, the lowest detection to the highest detection rate is RF < NB < LR < RFG < NBG < LRG. It is important to note that adding other parameters may affect the results differently. Moreover, we increase the training size from 160,000 to 500,000. The efficiency of the learning techniques reach up to 90% accuracy, but the training time is significantly taking longer time. However, In AIPRA, we have used 160,000 as a training size to have a result in a reasonable time.

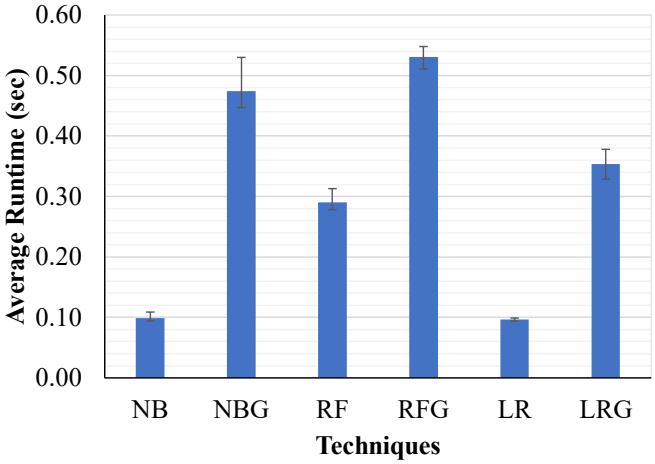

Fig. 5: The average runtime for the machine learning techniques.

2) Runtime Analysis: Fig. 5 shows the average time that the techniques take. The same experiment is run multiple times to analyze the runtime. The error bars show the minimum and maximum runtimes levels comparing to the average results. The runtime is only for technique and does not include the time for the rest of the programming parts such as obtaining geolocation. When only considering the features extracted from an IP address, the average runtime is almost 0.1 sec for NB and LR although LR has slightly lower runtime. On the other hand, RF has almost 0.3-sec runtime, which is more than three times of LR. With geolocation information, RF runtime (RFG) is the highest with 0.53 sec, while LR has the lowest runtime with 0.35 sec (LRG). While adding geolocation parameter increases runtime almost 0.25 sec in RF and LR, it increases almost 0.38 sec in NB. Therefore, the lowest runtime to highest runtime is LR < NB < RF < LRG < NBG < RFG.

E. Summary of Results

Based on the results, we make the following observations: (i) cross-checking system is better in terms of detection the malicious IPs in public databases but also decrease false positives, (ii) considering additional parameters with machine learning techniques to find IPs' reputations can affect the obtained results in a better way but increase runtime, and (iii) Ability in public databases and Logical Regression in machine learning techniques have higher detection rates.

IV. Conclusion and Future Works

The purpose of this paper is to analyze the efficiency of the public blacklist databases and machine learning techniques to detect reputation of the IPs and domains, then create a cross-checking system which automatically analyzes a number of reliable blacklist databases to find the reputation of the IPs and domains. If the information is not found by using cross-checking, the machine learning technique is applied to provide information. The developed IP Reputation Analysis can be found online at [16]. The results show that the developed analyzer is the most effective way comparing to the public databases. In our future works, private services such as Palo Alto will be investigated in terms of efficiency with a broader set of IPs. Moreover, for more effective results, an experiment with a variety of binary classification algorithms such as decision tree learning or using an artificial neural network will be tested. For this reason, more crawling will be required as well as new features to help identify a malicious IP address or FQDN. One such feature can be to take "snapshot" of a website or IP address and create a neural network to learn from the image of the website.

References

[1] K. AlRoum, A. Alolama, R. Kamel, M. Barachi, and M. Aldwairi, "Detecting malware domains: A cyber-threat alarm system," Mar. 27-28 2017.

[2] Anti-phishing protection in office 365. [Online]. Available: https://docs.microsoft.com/en-us/office365/securitycompliance/anti-phishing-protection

[3] H. Esquivel, A. Akella, and T. Mori, "On the effectiveness of ip reputation for spam filtering," in *Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010)*, Jan 2010.

[4] A. Renjan, K. P. Joshi, S. N. Narayanan, and A. Joshi, "DAbR: Dynamic attribute-based reputation scoring for malicious ip address detection," in *IEEE International Conference on Intelligence and Security Informatics (ISI)*, Nov 2018, pp. 64–69.

[5] A. Goswami, G. S. Parashari, and R. Gupta, "Evolutionary stability of reputation-based incentive mechanisms in p2p systems," *IEEE Communications Letters*, vol. 22, no. 2, pp. 268–271, Feb 2018.

[6] J. Yin, X. Cui, and K. Li, "A reputation-based resilient and recoverable p2p botnet," in *IEEE Second International Conference on Data Science in Cyberspace (DSC)*, June 2017, pp. 275–282.

[7] K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on dns," *Neural Computing and Applications*, vol. 28, no. 7, pp. 1541–1558, 2017.

[8] Virustotal database. [Online]. Available: https://www.virustotal.com

[9] AbuseIPDB. [Online]. Available: https://www.abuseipdb.com

[10] Apility. [Online]. Available: https://www.apility.io

[11] MyIP. [Online]. Available: https://www.myip.ms

[12] Shodan. [Online]. Available: https://www.shodan.io

[13] Censys. [Online]. Available: https://www.censys.com

[14] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security, 2010.

[15] J. Porenta and M. Ciglarič, "Empirical comparison of ip reputation databases," in *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, Perth, Australia, 2011, pp. 220–226.

[16] Automated ip reputation analyzer tool. [Online]. Available: http://ipreputation.herokuapp.com

[17] A. Thomas, "Rapid: Reputation based approach for improving intrusion detection effectiveness," in *Sixth International Conference on Information Assurance and Security*, Aug 2010, pp. 118–124.

[18] Y. Huang and P. Greve, "Large scale graph mining for web reputation inference," in *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[19] P. Lison and V. Mavroeidis, "Neural reputation models learned from passive dns data," in *IEEE International Conference on Big Data (Big Data)*, Dec 2017, pp. 3662–3671.

[20] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks," in *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, June 2015, pp. 403–414.

[21] S. Raj and R. Rajesh, "Descriptive analysis of hash table based intrusion detection systems," in *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, March 2016, pp. 233–240.

[22] Y. Tang, S. Krasser, P. Judge, and Y. Zhang, "Fast and effective spam sender detection with granular svm on highly imbalanced mail server behavior data," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Nov 2006, pp. 1–6.

[23] Z. Wang, M. Qin, M. Chen, C. Jia, and Y. Ma, "A learning evasive email-based p2p-like botnet," *China Communications*, vol. 15, no. 2, pp. 15–24, Feb 2018.

[24] J. Zhang, K. Jones, T. Song, H. Kang, and D. E. Brown, "Comparing unsupervised learning approaches to detect network intrusion using netflow data," in *Systems and Information Engineering Design Symposium (SIEDS)*, April 2017, pp. 122–127.

[25] H. H. Kilinc and U. Cagal, "A reputation based trust center model for cyber security," in *4th International Symposium on Digital Forensic and Security (ISDFS)*, April 2016, pp. 1–6.

[26] N. Wang and B. Wang, "A reputation-based method to secure inter-domain routing," in *IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, Nov 2013, pp. 1424–1429.

[27] M. A. Gosselin-Lavigne, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "A performance evaluation of hash functions for ip reputation lookup using bloom filters," in *10th International Conference on Availability, Reliability and Security*, Aug 2015, pp. 516–521.

[28] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[29] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] Scikit-learn. [Online]. Available: https://scikit-learn.org/

[31] Blacklists. [Online]. Available: https://www.dnsbl.info/dnsbl-list.php

[32] T. Menzies, S. Majumder, N. Balaji, K. Brey, and W. Fu, "500+ times faster than deep learning:(a case study exploring faster methods for text mining stackoverflow)," in *IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, 2018, pp. 554–563.