Future Generation Computer Systems 118 (2021) 124–141

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics

Nighat Usman [a], Saeeda Usman [b], Fazlullah Khan [c,d,*], Mian Ahmad Jan [e], Ahthasham Sajid [f], Mamoun Alazab [g], Paul Watters [h]

[a] *Department of Computer Sciences, Bahria University, Lahore, Pakistan*
[b] *Department of Electrical and Computer Engineering, COMSATS University Islamabad, Sahiwal, Pakistan*
[c] *Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City 758307, Viet Nam*
[d] *Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 758307, Viet Nam*
[e] *Department of Computer Science, Abdul Wali Khan University Mardan, KPK, Pakistan*
[f] *Department of Computer Science, Faculty of ICT, Balochistan University of Information Technology Engineering and Management Sciences, Quetta, Balochistan, Pakistan*
[g] *College of Engineering, IT and Environment, Charles Darwin University, NT, Australia*
[h] *School of Engineering and Mathematical Sciences, Latrobe University, Australia*

ARTICLE INFO

Article history:
Received 1 July 2020
Received in revised form 8 December 2020
Accepted 6 January 2021
Available online 11 January 2021

Keywords:
Cyber threat
Cyber security
Big data
Severity
Confidence level
Time to live
Machine learning
Essential selected features
Risk score

ABSTRACT

In the near future, objects have to connect with each other which can result in gathering private sensitive data and cause various security threats and cyber crimes. To prevent cyber crimes, novel cyber security techniques are required that can identify malicious Internet Protocol (IP) addresses before communication. One of the best techniques is the IP reputation system used for profiling the behavior of security threats to the cyber–physical system. Existing reputation systems do not perform well due to their high management cost, false-positive rate, consumption time, and considering very few data sources for claiming IP address reputation. To overcome the aforementioned issues, we have proposed a novel hybrid approach based on Dynamic Malware Analysis, Cyber Threat Intelligence, Machine Learning (ML), and Data Forensics. Using the concept of big data forensics, IP reputation is predicted in its pre-acceptance stage and its associated zero-day attacks are categorized via behavioral analysis by applying the Decision Tree (DT) technique. The proposed approach highlights the big data forensic issues and computes severity, risk score along with assessing the confidence and lifespan simultaneously. The proposed system is evaluated in two ways; first, we compare the ML techniques to attain the best F-measure, precision and recall scores, and then we compare the entire reputation system with the existing reputation systems. Our proposed framework is not only cross checked with external sources but also able to reduce the security issues which were neglected by existing outdated reputation engines.

1. Introduction

Smart devices have been turning out at quick speeds throughout the last decade. The Internet of Things (IoT) is an evolving innovation that enables the ability to link things/objects with the computerized world for transmission of data [1]. However, the majority of these IoT objects can be easily hacked and compromised [2]. Due to which the security of IoT has turned into a demanding concern. The danger exposed to the smart devices needs to be addressed [3]. The battle between security experts and malware designers is an endless fight. The behavior of malware changes as fast as advancement develops in the domain of securing things. Modern research emphasizes on the development of things due to which the patterns of malwares are also evolving. In order to detect and identify such malwares the Machine Learning (ML) methods are utilized. To stay aware of malwares, security experts and specialists need to persistently expand their cyber defenses. One fundamental component is a maximum secure system at the endpoints. Endpoint defense offers a group of security plans for instance, firewall, anti-spam, email security, URL filtering and sandboxing. Nowadays ML plays a very significant role in cybersecurity for anomaly detection. The

[*] Corresponding author at: Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City 758307, Viet Nam.

E-mail addresses: nighat.bulc@bahria.edu.pk (N. Usman), saeeda@cuisahiwal.edu.pk (S. Usman), fazlullah@tdtu.edu.vn (F. Khan), mianjan@awkum.edu.pk (M.A. Jan), ahthasham.sajid@buitms.edu.pk (A. Sajid), Mamoun.Alazab@cdu.edu.au (M. Alazab), p.watters@latrobe.edu.au (P. Watters).

https://doi.org/10.1016/j.future.2021.01.004
0167-739X/© 2021 Elsevier B.V. All rights reserved.

Internet Protocol (IP) reputation and confidence play a fundamental role in web applications while creating faithful relationships among the number of organizations for their common interest. Several approaches such as antivirus, websites signifying blacklist IPs, and reputation systems have fascinated organizations as promising ways for cybersecurity [4]. Several approaches (such as signature-based techniques [5], behavioral-based techniques [6], anomaly-based techniques [7], etc.,) are employed to be used as the base for reputation systems. However, according to [8] and [9] behavioral-based approach is more efficient than the signature and anomaly-based techniques. In a behavioral-based approach, an entity's behavior is examined to score its reputation.

By deploying reputation systems, organizations can protect their machines or network from malicious IP addresses which are having a history of cybercrimes. In literature, several ML techniques such as; Mini Batch K-means (MBK) clustering [10], Naive Bayes (NB) [11], Support Vector Machine (SVM) [12] and Decision Tree (DT) [13], techniques are used for predicting anonymous behavior in big data [14–16]. However, there are some limitations in the existing reputation systems, for instance, they often fail to detect a zero-day anomaly, may rely only on internal sources, may generate a high false-positive rate, may be outdated due to high management cost. The reason behind the discovered issues is the lack of data sources. On the other hand, existing reputation systems depend on various antivirus, ML techniques, and blacklisted IPs. However, these approaches do not accurately categorize zero-day attacks and are not updated timely due to its high management cost. Our motivation is to classify zero-day attacks, reduce false alarm rates, update suspicious IP addresses, reduce management cost, and generate the forensics. Malicious files are not always the same, they can be variants of existing malware families or maybe zero-day malware that are unknown to the world. To obtain the optimal results a novel reputation system is required.

Due to the aforementioned shortcomings [17,18] notes that existing reputation systems are not preferred. To overcome the aforementioned deficiencies, we proposed a framework "Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics" (IP Rep. - FDA). We have considered big data analytics to expose the unknown hidden patterns, mysterious correlations, customer preferences, and other significant knowledge, to assist organizations in making appropriate business decisions. To tackle IP addresses that are associated with such kinds of suspicious files, several ML techniques are applied to the behavior of suspicious files. While considering reputation systems, ML is used to detect patterns in data. ML techniques are categorized as supervised (Classification) and unsupervised (Clustering) learning. Among the aforementioned listed techniques, we suggest that the DT technique is considered as an appropriate classifier for categorizing new malware due to its nice probabilistic approach than SVM. We argue that SVM works appropriately for a dataset containing fewer features but lags when several essential features are improved in number. NB becomes biased. DT's computational cost is less than MBK and induces a low false-positive rate than other listed techniques [14–16]. We argue that to repute an IP address, its risk, confidence, lifespan as well as its associated behavior are required to be analyzed simultaneously. While all these factors are not considered simultaneously by existing approaches due to which the existing approaches lag in uncovering the hidden patterns. The confidence level is the computed value that can assure the maliciousness level of an IP; whereas, lifespan forecast the time to live (TTL) of spotted IP. For scoring risk factors, the severity of associated malware is required to be computed. For forensics, malware associated with the spotted IPs is dynamically analyzed in the Cuckoo sandbox to capture the file, registry, and network-based behavior of malware [19,20]. To categorize a behavior of zero-day malware, a DT classifier is applied to the samples of malware. Generated reports are analyzed to determine the severity of a threat and combined with the results of cross-referencing and the number of hits for risk estimation.

The evaluation of the proposed framework relies on two ways: first, we compare the number of ML techniques on two of our datasets. $Dataset_1$ consists of records containing only network relevant information of binaries. Like $Dataset_1$, $Dataset_2$ also includes network-related information but it also contains file system and registry-based information. Next, we compared the proposed framework with the existing systems to illustrate the limitations of existing approaches. The key contributions of our paper are listed below:

1. We have classified malware families using several ML techniques, where DT technique performs best due to its comprehensive analysis nature and generates promising results. The DT technique shows 93.5% accurate predictions. To demonstrate the malicious behavior of an IP address, we dynamically analyze the malware sample associated with the IP address. For every occurrence of an IP address the reputation is computed at run time which shows the most updated information
2. The proposed framework is able to detect zero-day anomaly by analyzing the behavior of malwares in the Cuckoo sandbox. The proposed framework is designed in such a way that it will alarm the user with the most updated information without introducing any extra management cost. The framework extends the existing reputation engine's approach by considering the external sources as well;
3. The reputation of a particular IP address is computed again and again upon its re-occurrence in the system and is updated within no time which reduces the false alarm rate regarding maliciousness of IP address; The behavior of any two IP addresses can be different, one can be more risky with respect to the other due to which the weighted risk score is computed. The IP address can re-attempt to cause the damage due to which weight is assigned to the IP address for computing the updated risk score;
4. The proposed study aims to build the confidence level by utilizing both internal and external sources to assure the maliciousness level of an IP address; We forecast a TTL period for the spotted IP address which further improves the assurance level; Moreover, the proposed framework is also able to reduce the false alarm rate as we are considering both internal and external sources.

The structure of the rest of the paper is as follows: Section 2 comprises a related work that covers the relevant research in the field of the reputation system, blacklists, sandboxes comparison. Section 3 illustrates the proposed system. After that, results and discussions are enclosed in Section 4. Finally, we conclude our research work in Section 5.

2. Related work

Existing systems provide multiple possible solutions to the firms; however, due to several shortcomings, reputation is under construction for every kind of object. Few parameters (such as; severity, risk, confidence, and lifespan) are examined that demonstrates the comprehensive study of existing approaches.

2.1. Entity reputation

A confidence score for the IP address is calculated based on pre-attack information [21]. Pre-attack information about IPs is collected from multiple sources such as; net flow information, blacklisted IPs information, number of connections, activation time, location of IP addresses, and so on. A system generates a confidence score based on this extracted knowledge. According to [22], the severity score depends upon the number of occurrences and history of an incident for an IP address in network relevant datasets. Moreover, data sources that indicate the weighing factor for an IP address also contributes to the computation of severity score for a range of IP addresses. An approach is proposed by [10], that computes the approximate risk value for an entity domain. Unsupervised learning is used to make a distinction between benign and malicious behavior. An open-source Cuckoo sandbox is used for collecting the processes communication. System calls for processes' communication with endpoints of the network were captured for learning the behavior. Behavior-based approaches are more appropriate for detecting hidden behaviors of malware samples than signature-based techniques [5,6]. Detecting domain names that were associated with C&C servers are of high significance. Malignant software' uses these domains which were tested by domain generation algorithms [23]. In general, risk computing measures mark hosts as blacklisted for an association with specific malware or block specific IP address on communication with C&C server [24].

The risk score for a web user is determined in [25]. The observed actions are classified and assigned a risk score based on the activities performed by the user. A criterion is defined for computing the risk formula, which includes some risk computing and weighing factors. Compromised machines with a range of security products were located in the organization [26]. With the help of security endpoints, an attack is automatically detected and a reputation service is enabled that reports information regarding associated URLs and IP addresses.

In [27], a feature modeling with data mining technique is proposed that identifies vulnerabilities to the websites and assigns a quantitative score to the vulnerability. A tool named as malicious code finder is used to detect malicious codes hosted by suspicious websites. This tool scan a number of suspicious websites and generates logs which were used as an input for the data mining step. By analyzing the logs a severity score is assigned to the website hosting malicious content. Severity includes; history that covers detection rate and sites linked to the malicious content distribution site. The higher the severity means risk level is high. A scanner for detecting vulnerability known as Nikto is used to check whether a certain website is vulnerable to threats or not. After applying Nikto, feature modeling is performed to detect any vulnerability. The paper focuses only on a threat level known as risk score and did not consider any other factor for assuring the maliciousness of threat.

One of the most significant methods for making an Intrusion Detection System (IDS) is to practice ML techniques [28]. In this paper, the author used the IDS for directly monitoring the network oriented logs where every operation is either benign or malevolent. The aim is to sense and notify the network controllers when attack occurs. By discovering any malignant activity the IDS is supposed to immediately block the link. The behavior of concerned activity is not analyzed which can generate high false alarm rates.

The motive of the authors in [35] is to examine malignant websites using the features of URL entity. The study claims that there exist some features in the network address which reflects some time change factor. On the basis of this time change factor the malignant websites are classified, however they did not dynamically analyze any malware sample.

In [36] the authors proposed a method for predicting the zero-day IP addresses which were not blacklisted before by any reputation engine. The goal of the study is to discover those IP addresses which will be used by the attacker in the near future. The prediction is based on the reports generated by the cyber security intelligence team. The Top Level Domain (TLD) files for .com and .net extension are used. The TLD list helps in identifying those IP addresses which allocates several domains possessed by different organizations. The IP address that hosts several websites and exists in GT Malware and Phishtank in past N days, where N is the time-window size is more likely to be the one that can become vulnerable in the near future.

In [37] the Automated IP Reputation Analyzer Tool (AIPRA) is developed, which automatically scrutinizes a number of databases that contain blacklisted IP addresses. Number of ML techniques are applied in the proposed paper and a parameter geolocation is considered to identify the malignant IP address. The geolocation is integrated with the ML techniques. The authors claim that due to ML techniques the generated results are different from several available databases that contain blacklisted IP addresses.

2.2. Malware sample inspection

The main goal of [38] is to analyze malware and report against their activities. Generally, two ways aye followed for analyzing malware samples. One way is to perform static analysis which helps in discovering faults in memory corruption. But result in obfuscation related issues and loss of information, that is why it is not preferred [39]. Due to the inadequacies of static analysis [12], dynamic analysis is favored and followed in this paper to demonstrate a complete structure of the activities of the sample. Usually, a sandbox or virtual machine is used to retain the logs in a controlled environment for the performed activity. Various open-source and commercial sandboxes (such as Cuckoo Sandbox [40], Threat-Expert [41], Norman Sandbox [42], Anubis [43], Joe Sandbox Document Analyzer [44] and so on) in the form of service can be accessed from the internet. In this paper, the cuckoo sandbox is used to present the main features of the binary executable file. The binary file execution time depends upon the suspicious behavior of binary files [45]. These features are then clustered to identify the hidden processes of malware.

A real-time reporting system is proposed by [20], which combines the number of analysis tools and methods in a single architecture. The system automatically reports about the activities performed by the zero-day attack by integrating manual, static, and dynamic analysis in a single component. According to [46], it is forecasted by the security professionals that daily 70,000 variants of malware are launched. Organizations or enterprises try to secure themselves from zero-day attacks by moving from signature-based technique [47–49] towards behavioral-based [50–53] and anomaly-based approaches [54]. According to [55], although various kinds of reputation systems are available but still malware is considered as one of the major threats to the organizations. A comparison table is illustrated in Table 1, in which the number of features is examined and the comprehensive study of existing approaches is carried out.

Several shortcomings are discovered in the existing reputation systems shown in Table 2. These limitations are of high significance and need to be addressed promptly via trusted remedial action. To overcome these listed shortcomings, an alert based automated reputation system is proposed to repute an IP address and score zero-day anomalies in a pre-acceptance manner. For computing IP reputation score, the behavior of IPs' associated malware (known attacks or zero-day attacks) is dynamically analyzed in the sandbox, malwares are well categorized by DT classifier and IPs are cross-referenced (historical examination)

Table 1
A comparative analysis of existing reputation approaches.

Reputation	Input type	Analysis	Mode of operation	Strategy	ML technique	Parameters	Limitations	Ref. no.
DNS	Malign and benign	Dynamic	Pre-acc.	Trained classifier	Mini batch k-means	Risk estimated	MBK is not applicable for large datasets	[10]
DNS	Malign and benign	Static and dynamic	Pre-acc.	History analysis	Classification and clustering	Positive and negative reputation score	Consumes more time and training data	[29]
Web server	Malign	Static	Pre-acc.	Trained classifier	Supervised, Bag of words	Severity of threat	Order required by web is missing, no accuracy measurement	[4,30]
Web server	Malign	Dynamic	Pre-acc.	Fuzzing based triggering technique	No ML technique applied	Malware identification performance	No accuracy measurement criteria, not applicable for long lasting communication	[31]
User	Malign and benign	Static	Pre-acc.	Classifying user actions, behavioral analysis	No ML technique applied	Risk assessment., report generation and alerts	High false positive rate	[25]
User	Malign and benign	Static	Post-acc.	Referring reputation system	No ML technique applied	Alerts generation	High false positive rate, slow	[26]
User	Malign and benign	Static	Post-acc.	Kullback–Leibler divergence, Spearman's rank correlation coefficient	No ML technique applied	Threat severity, negative reputation	No comparative analysis	[32]
User	Malign and benign	Static	Post-acc.	Behavioral analysis	Logistic Regression, DT, SVM, and Bayesian classifiers	Risk prediction	High false positive rate	[13]
User	Malign and benign	Static	Pre-acc.	Behavioral blacklisting	Classification and clustering	Malign pattern	Not applicable for large size of dataset	[33]
Files	Malign and benign	Dynamic	Post-acc.	Comparison based approach	Supervised, RF, DT, KNN, SVM	Malware detection, better supervised technique for dynamic analysis	Old approach, no score for detected malwares	[12]
Files	Malign and benign	Static and dynamic	Pre-acc.	Rule based comparison and history analysis	DT	Malware prediction	No comparative analysis, no risk score for entity	[11]
IP	Malign and benign	Static	Pre-acc.	Threshold value specified and history	No ML technique applied	Confidence score and ranking of blacklisted IPs	Flow based analysis consumes more time	[21]
IP	Malign	Static	Pre-acc.	Addition of weighing factor	No ML technique applied	Risk assessment	High false positive rate, slow	[22]
IP	Malign and benign	Static	Pre-acc.	Reverse DNS lookup and keywords matching criteria	No ML technique applied	Risk assessment	No class for detected malign IPs to assign scores	[34]

Table 2
Limitations in existing reputation systems.

i.	Consumes more time
ii.	Outdated websites signifying blacklisted IPs
iii.	Computational cost is too high
iv.	High false positive rate
v.	Overlooked past behavior
vi.	Computational cost is too high
vii.	Lack of confidence score
viii.	Lack of lifespan

referenced to various scanners (such as VirusTotal [56], OTX [57], MyIP [58]) for IP history.

The number of aforementioned limitations are overwhelmed by the proposed framework. Several ML techniques are applied to discover the best technique for getting the promising results. The framework is able to reduce the false alarm rate by computing the severity, confidence and lifespan of each IP address that reappears. The framework is compared with various existing reputation engines which shows that the projected framework is able to identify malwares and detect zero-day anomalies accurately. For big data analytics and cyber-crime detection, the proposed solution is capable of achieving successful results.

3. Intelligent Dynamic Malware Detection using ML in IP Reputation for Forensics Data Analytics (IP Rep. - FDA)

For a well-reputed system, we proposed a hybrid approach known as "Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics". We have deployed two honeypots working as the Intrusion Detection System (IDS) for capturing the malicious activities. An adversary endeavors to ping and download suspicious binary files; this entire activity is retained in the database. We have also downloaded a number of suspicious binaries from several web links to collect the maximum samples and claim the correct reputation regarding the plurality of IP addresses. For gaining knowledge regarding IP addresses, we perform forensics on the malware. We dynamically analyze the malware samples in a sandbox named Cuckoo to trace the behavior of binaries. The binaries are executed in Cuckoo sandbox dynamically which executes a particular binary file entirely. The execution time depends upon the binary file procedures. The cuckoo sandbox automatically classifies the File system, Registry, and Network activities. Information regarding files, processes, registry, and the network is logged and retained in a file. Desired features are extracted from the output file to avoid redundancy and overhead. To compute the degree of danger posed by these binaries, we have analyzed the file system, registry, and network-based activities. From the extracted results, we get some labeled and unlabeled binaries. The report having a name for malware basically depicts the malwares which are known to the antivirus software, while the one whose name is unknown is considered as zero-day malware. In order to label them, we classify them by applying a ML technique.

We have designed a system that grants quantitative knowledge regarding four parameters; Severity of threat (degree of danger) [4], Risk [31], Confidence level, and Lifespan of an IP. These 4 parameters play a very significant role in computing the reputation of an IP address. A conceptual model of our IP reputation framework is illustrated in Fig. 1. There are ten main phases through which the data needs to be passed and examined. Each phase is discussed one after one below:

3.1. Preprocessing phase

In the preprocessing phase, we have deployed two honeypots for the detection of malicious IPs and their attacks. During the activation period of an IP address, we extract a number of features which are; an IP address, its associated malicious binary file, its activation time, and its number of attempts. All these features collectively aid us in computing the reputation for an IP address. We have also taken several malicious binary samples from web links to reflect the big data analytics.

3.2. Malware analysis

Malware analysis is the practice followed to identify the source and principle objective of a certain binary file (such as; virus, backdoor, worm, or trojan horse). For analyzing a malware sample, generally, two ways are followed which are briefly discussed below:

3.2.1. Static analysis

Static analysis is also known as code analysis, usually performed for detecting the suspicious level by splitting the various functions of the particular malware sample. The code for the various extracted functions is analyzed statically without executing the malware sample. However, static analysis is not preferred as tools lack in performing analysis when the code is obfuscated or the malware sample is wrapped. Thus, this technique is not considered as an optimum solution for analyzing the malware files [59].

3.2.2. Dynamic analysis

The dynamic analysis is also known as the behavioral analysis is usually performed to examine the behavior of the spotted binary file during its execution process on the host machine. During the processing of instructions, multiple debuggers are used to observe the activities and impact on the host machine by the malware [59]. For avoiding any kind of mishaps we used a sandbox for dynamically analyzing the binaries. In the proposed paper, we have used a sandbox named Cuckoo for dynamically analyzing the binaries samples. In order to identify forensics, the executable file as input is sent to the cuckoo which executes the file and returns its activities in an HTML format. The malware sample is analyzed to observe the behavior intended to perform. According to the requirements of our system, we collect several features which are listed in Table 3.

Table 3
Selected features for the IP reputation framework.

i.	source.target.md5Hash
ii.	source.virustotal.results.total
iii.	source.virustotal.results.positives
iv.	source.malscore
v.	source.dropped.guestpaths
vi.	source.network.udp
vii.	source.network.http
viii.	source.network.tcp
ix.	source.network.icmp
x.	source.network.smtp
xi.	source.network.hosts
xii.	source.network.dns
xiii.	source.network.domains
xiv.	source.network.irc
xv.	source.behavior.summary.files.write.files
xvi.	source.behavior.summary.files.delete.files
xvii.	source.behavior.summary.keys.write.keys
xviii.	source.behavior.summary.keys.delete.keys
xix.	source.behavior.summary.mutexes
xx.	source.malfamily

After collecting the values for the mentioned feature, we have observed that in some of the instances, malfamily is null which means that behavior exists for the malware but its name is unknown to the world. For labeling such kinds of malwares, the proposed system labels these samples by applying a ML technique on the sample set of malwares. That specific kind of malware can be a variant of any other malware family or maybe a zero-day malware. We call it zero-day as for this malware, the score in the column of source.virustotal.results.positives is 0 which means that none of the antivirus software detects this specific malware. That is why it is unlabeled in a column of malfamily. The unlabeled instances are anomalies, which means that they are unknown and did not match any existing signatures. To label these kinds of instances we apply the ML technique on the input data. For computing the similarity of these anomalous samples with the known samples we have applied a DT technique. By doing so, we have achieved labeled records and categorized identified anomalies in this way.

3.3. Malware family classification

In this phase, data is collected from the malware analysis phase. Some of the instances have the labeled malware family while some of them are unlabeled. To label these kinds of instances, we apply ML technique on the input data. For computing the similarity of these anomalous samples with the known samples, we have applied a DT technique. By doing so, we achieve labeled records. The identified anomalies are categorized in this way.

3.4. Severity of malware

Once the malwares are assigned labels, the next step is to assign severity score to these malwares. Severity score depends upon the methodology of Kill Chain introduced in 2011 at Lockheed-Martin Corporation by [60]. The kill chain term was initially used by a martial force that demonstrates the formation of an attack. It identifies the target, force to attack the target, the procedure for carrying out the task, and lastly achieving the aim of the task [61]. Kill chain methodology based on the steps

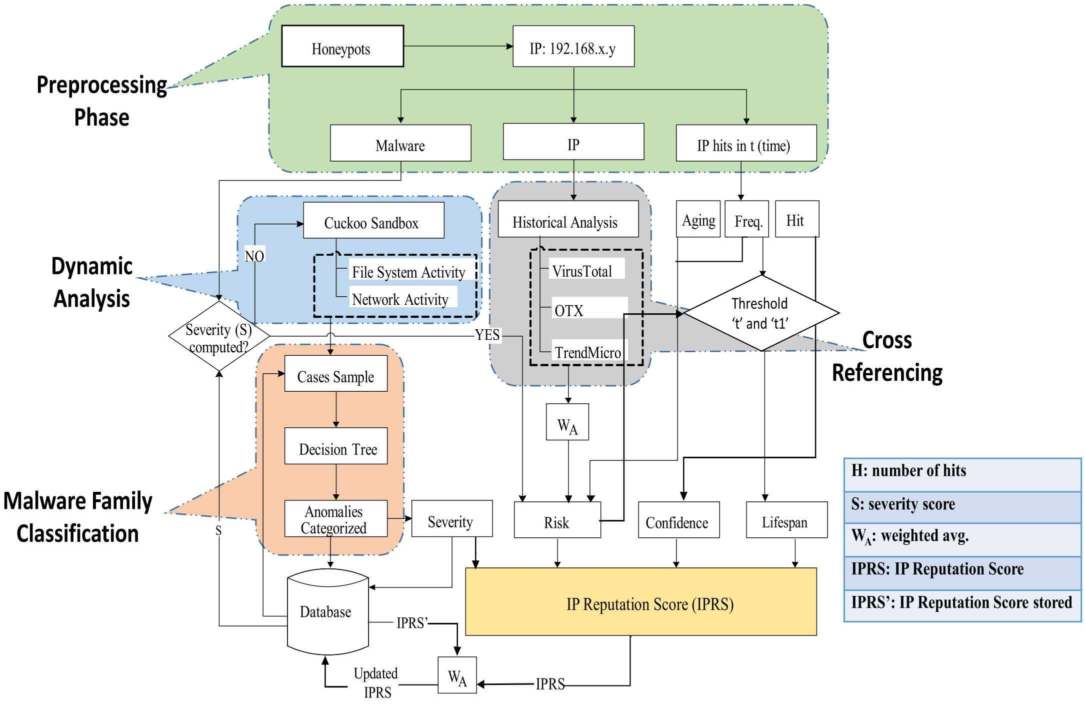

Fig. 1. Proposed IP reputation system.

listed in Fig. 2. The algorithm and flowchart for the severity calculation are shown in Algorithm 1. Based on the kill chain methodology, the threat of danger is calculated. By doing so, we get the severity of a malware sample. Moreover, we assign a weight to the computed severity factor as it contributes more than the other listed factors.

3.5. Observed behavior

In this phase, we monitor the behavior of IP addresses. The way an IP address is attacking and exploiting the targeted network. We observe the IP address by two means listed below:

3.5.1. Frequent behavior

An IP's frequent behavior is observed by keeping its time and malware sample. Frequency indicates the number of times an IP is attacking the target's network within a certain time. Like severity, weight is also assigned to the frequency factor but less than severity according to the priority.

3.5.2. Aging

Represents the time difference between the two occurrences of an IP address. Time ranges are defined to monitor the gap between two occurrences of an IP address. Like severity and frequency, aging is also assigned a weight but less than the severity and frequency factors. The algorithm for the computation of aging is shown in Algorithm 2.

Big data analytics applications contain data from both internal systems and external sources. Therefore, after computing the values from internal systems information, the next phase is to attain information regarding detected IP addresses from external sources.

3.6. Historical analysis

We examine an IP address historically for three purposes; the first reason is to compute the risk score, the second purpose is to compute the confidence level, whereas, our third intention is to consider more data sources to provide knowledge-based on big data. For getting the historical information regarding an IP address, we have taken aid from three external sources which are VirusTotal, OTX (Open Threat Exchange), and MyIP. These 3 external sources contribute to computing the Risk value and Confidence level for the detected IP addresses. The maximum weight is assigned to the virus total than the other two external sources as it consists of 56 antivirus software that triggers an alarm on detecting the malicious IP address. After that, OTX is assigned a second highest weight and in the end, MyIP is assigned some weight according to the priority. The purpose of these external parties is to be more confident at the risky level of an IP address.

$$Ext.sources = [(VT * w'_1) + (OTX * w'_2) + (MyIP * w'_3)] \qquad (1)$$

where, VT is used to represent VirusTotal; w'_1, w'_2 and w'_3 represents weight assigned to the external sources based on the priority; VT, OTX, and MYIP are the external sources that result in the binary form either 1 or 0.

In this case, 1 indicates the existence of an IP address in the database of an external party whereas, 0 shows that a particular IP address is not listed in their databases. "Not listed" does not mean that an IP address is white-listed or not blacklisted. We have discovered a case where an IP address needs to be blacklisted as it downloads a malicious binary several times but was not listed in the external parties' databases so we cannot call it a secure IP address.

3.7. Risk score

After computing the external sources' information, the next step is to aggregate all the previously calculated scores and compute the risk score for an IP address. We have assigned more weight to the internal sources information than external sources information as we have observed that for most of the IPs, information is missing in the databases of external sources. Therefore,

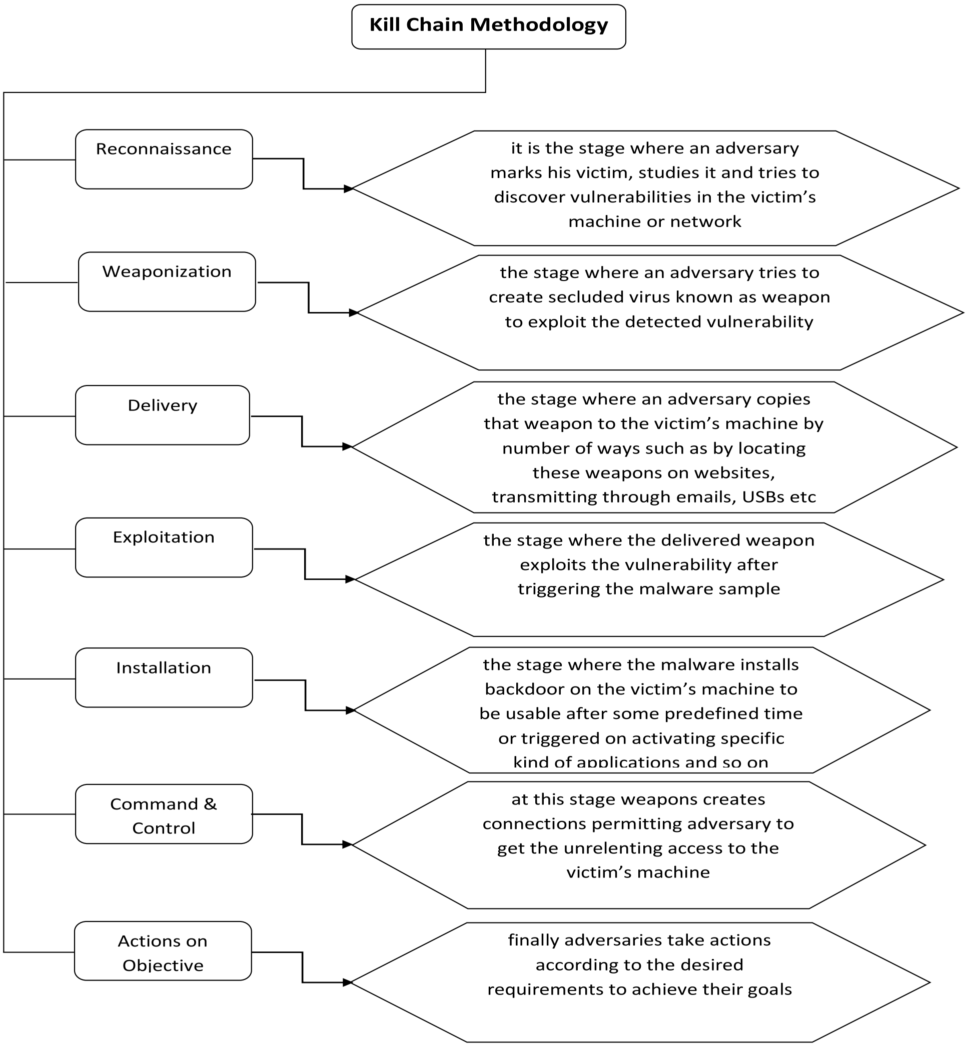
Kill Chain Methodology
Reconnaissance
it is the stage where an adversary marks his victim, studies it and tries to discover vulnerabilities in the victim's machine or network
Weaponization
the stage where an adversary tries to create secluded virus known as weapon to exploit the detected vulnerability
Delivery
the stage where an adversary copies that weapon to the victim's machine by number of ways such as by locating these weapons on websites, transmitting through emails, USBs etc
Exploitation
the stage where the delivered weapon exploits the vulnerability after triggering the malware sample
Installation
the stage where the malware installs backdoor on the victim's machine to be usable after some predefined time or triggered on activating specific kind of applications and so on
Command & Control
at this stage weapons creates connections permitting adversary to get the unrelenting access to the victim's machine
Actions on Objective
finally adversaries take actions according to the desired requirements to achieve their goals

Fig. 2. Kill chain methodology steps.

we assigned α and β weight to the internal and external sources respectively. For that we have,

$$RiskScore(R.S) = \frac{\alpha(Int.sources) + \beta(Ext.sources)}{(\alpha + \beta)} \tag{2}$$

where,

$$Int.sources = [(sev * w_1) + (freq * w_2) + (ag * w_3)] \tag{3}$$

$$Ext.sources = [(VT * (w'_1)) + (OTX * (w'_2)) + (MyIP * (w'_3))] \tag{4}$$

$$R.S = \frac{\alpha(S * w_1) + (F * w_2) + (A * w_3) + \beta(VT * w1') + (OTX * w2') + (MyIP * w3')}{(\alpha + \beta)} \tag{5}$$

Here, *sev* represents the severity of identified malware; *freq* represents frequency (number of attempts/occurrences) of detected IP address; *ag* represents aging of detected IP address; $w_1, w_2, w_3, w'_1, w'_2$ and w'_3 represents weight assigned to the internal and external sources information based on the priority.

With the help of AHP tool we have computed the following values:

$\alpha = 87.5$;

$\beta = 12.5$;

$VT = [0, 1]$;

$OTX = [0, 1]$;

$MyIP = [0, 1]$;

$w_1 = 0.625$

$w_2 = 0.238$

$w_3 = 0.419$

$w'_1 = 0.558$

$w'_2 = 0.32$

$w'_3 = 0.122$

This is how; we compute IP's risk score on the first time detection. The algorithm for the computation of risk score is shown in Algorithm 3. A risk score shows the risky level of an IP address. By having the computed risk score, an organization can decide where to communicate and what kind of privileges need to be provided to the users.

3.8. Confidence level

After computing the severity and risk score, we assess the confidence level for our reputation system. A confidence level is computed by taking information from external sources to reflect assurance based on big data. The more IP is listed in the external sources databases the more confident we are about the bad

Algorithm 1 Computation of Severity score for Malware

```
Input: A Malicious File $MF$
Output: A severity score based on malware behavior.
procedure
    find $MF$ in database
    if $MF$ exists then
        $S = MF.S$
        return $S$
    endif
    else
    Sends $MF$ to cuckoo
    Analyze Behavior
    Apply Decision Tree
    Store $case$ in $CaseTable$
    if Objective Fulfilled then                ▷ Actions on Objectives
        $S = x$                                ▷ Max Weight
        Store $S$ in $SeverityTable$
        return $S$
    endif
    else
    if Network Communication found then        ▷ C&C
        $S = y$                                ▷ Second Max Weight
        Store $S$ in $SeverityTable$
        return $S$
    endif
    else
    if Mutex created then                      ▷ Install
        $S = z$                                ▷ Third Max Weight
        Store $S$ in $SeverityTable$
        return $S$
    endif
    else
    if file-based Malicious Activity then      ▷ Exploit
        $S = \alpha$                           ▷ Fourth Weight
        Store $S$ in $SeverityTable$
        return $S$
    endif
    else
    $S = d$                                    ▷ Default value
    Store $S$ in $SeverityTable$
    return $S$
```

behavior of an IP address. We have assigned an equal percentage to every source of information.

For computing the confidence level we use:

$$ConfidenceLevel(CL) = [hit + hit * (VT + OTX + MyIP)] \quad (6)$$

where,
$hit = 25\%$;
$VT = [0, 1]$;
$OTX = [0, 1]$;
$MyIP = [0, 1]$

For instance, if an IP is listed in the database of VT, then VT will be equal to 1 otherwise 0. The same criteria are also applied to the other two external sources. This is how; we assure the reputation for an IP address as shown in Algorithm 4.

3.9. Lifespan

Moreover, we estimate the lifespan of an IP address based on the IBM QRadar SIEM engine. QRadar can analyze logs at run time and identify malicious behaviors. It is also capable of assessing the lifespan for an IP address. QRadar considers three cases for assessing the lifespan which is discussed below:

Algorithm 2 Computation of Aging for monitoring timeslap between two occurrences of an IP address

```
Input:
a) current time $T_c$
b) previous time $T_p$
Output: Aging associated with an IP.
procedure
    if $(T_c - T_p) < 7 days$ then
        Aging=$a$
    endif
    else
    if $(T_c - T_p) > 7$ && $(T_c - T_p) <= 21$ then
        Aging=$b$
    endif
    if $(T_c - T_p) > 21$ && $(T_c - T_p) <= 42$ then
        Aging=$c$
    endif
    endif
    if $(T_c - T_p) > 42$ && $(T_c - T_p) <= 70$ then
        Aging=$d$
    endif
    endif
    if $(T_c - T_p) > 70$ && $(T_c - T_p) <= 91$ then
        Aging=$e$
    endif
    else
    Aging=$x$                                  ▷ x is a default value
```

Case I: if an IP address has been detected more than a specified threshold 't' and has risk score more than the specified threshold 't_1' then that specific IP will last more than 90 days. This particular IP will be assigned to the Long TTL class. The threshold 't' and 't_1' are computed using the AHP tool. Here, the value for 't' is 0.6 while 't_1' is 0.4. More weightage is assigned to the risk score than frequency due to the maximum contribution of the risk score in categorizing a suspicious IP address.

Mathematically:

$$Lifespan = [freq > t \quad \&\& \quad riskscore > t_1] \quad (7)$$

$$Lifespan > 90 \text{ days} \quad (8)$$

Case II: an IP address has been detected more than a specified threshold 't' but has risk score less than the specified threshold 't_1' or vice versa then that specific IP can last till 90 days. This particular IP will be assigned to the Medium TTL class.

Mathematically:

$$Lifespan = [freq > t \quad \&\& \quad riskscore < t_1] \quad (9)$$

or

$$Lifespan = [freq < t \quad \&\& \quad riskscore > t_1] \quad (10)$$

$$Lifespan = 30 \quad to \quad 90 \text{ days} \quad (11)$$

Case III: an IP address has been detected less than a specified threshold 't' and has risk score also less than the specified threshold 't_1' then that specific IP will last more till 30 days. This particular IP will be assigned to the Short TTL class.

Mathematically:

$$Lifespan = [freq < t \quad \&\& \quad riskscore < t_1] \quad (12)$$

$$Lifespan = 01 \quad to \quad 30 \text{ days} \quad (13)$$

We also forecast the lifespan of an IP address according to the QRadar SIEM criteria. By following the predefined criteria our system is able to predict the TTL class in which an IP address can fall. The algorithm and flowchart for the computation of IP's TTL are shown in Algorithm 5.

All the computed parameters such as severity, risk score, confidence level and lifespan show us the reputation of an IP address.

3.10. Updated risk score

To make our reputation system efficient, we do not compute severity scores again and again, for instance; if it is previously calculated we store it in the database and reuse it when required. While concerning risk score, we recalculate a risk score, although the severity is the same if its associated binary is the same as the previous one but the reason behind recalculating risk score is that to test an IP address again in the external sources to identify its updated status. We apply a concept of weighted average to aggregate the previously calculated risk score. The way risk score is calculated as illustrated here

$$UpdatedRiskScore = [(0.3) * (prev_{R.S}) + (0.7) * (new_{R.S})] \tag{14}$$

where R.S is representing Risk Score.

The purpose behind the weighted average is to be familiar with the nature of an IP address as we have discovered that an IP address with the same malware can attack several times, therefore we need to upgrade its risk score to show its risky level. We assign less weight to previously calculated risk scores than the new risk score as the new score is of more significance than the previous one. Likewise, the Confidence level is also updated to present an updated guaranteed status. The detailed framework of the proposed solution is shown in Fig. 3.

The evaluation of the proposed IP reputation framework relies on two factors; first is to compare the ML approaches which are intended for achieving an optimum solution while categorizing malware families. Next, we compare the IP reputation framework with the existing reputation systems. By doing so, the limitations of the existing approaches are shown. Zero-day attacks are automatically categorized through DT algorithm and thus false alarm rate (actually yes, predicting no) is reduced without introducing any management cost.

Algorithm 3 Computation of Risk score for discovering threat level from an IP address

```
Input:
a) IP
b) A malicious file $MF$
Output: Risk associated with an IP.
procedure
    send $IP$ to $virusTotal$
    if $IP$ exists then
        $VT = 1$
    endif
    else
    $VT = 0$
    send $IP$ to $OpenThreatExchange$
    if $IP$ exists then
        $OTX = 1$
    endif
    else
    $OTX = 0$
    send $IP$ to $MyIP$
    if $IP$ exists then
        $MyI = 1$
    endif
    else
    $MyI = 0$
    calculate $Frequency$
    F= No of Occurrence/Unit Time
    calculate $Aging$
    A= $Aging(T_c, T_p)$
    calculate Severity
    S= $Severity(MF)$
    Risk=
        $\frac{\alpha(S*w_1)+(F*w_2)+(A*w_3)+\beta(VT*w1')+(OTX*w2')+(MyI*w3')}{(\alpha+\beta)}$
```

Algorithm 4 Computation of Confidence level for reflecting behavior of an IP address

```
Input: $IP$
Output: Confidence Score.
procedure
    send $IP$ to $virusTotal$
    if $IP$ exists then
        $VT = 1$
    endif
    else
    $VT = 0$
    send $IP$ to $OpenThreatExchange$
    if $IP$ exists then
        $OTX = 1$
    endif
    else
    $OTX = 0$
    send $IP$ to $MyIP$
    if $IP$ exists then
        $MyI = 1$
    endif
    else
    $MyI = 0$
    Confidence=
    $(Hit * Internal) + VT * Hit_V + OTX * Hit_O + MyIP * Hit_M$
    return Confidence
```

Algorithm 5 Computation of IP Lifespan for predicting its reoccurrence time range

```
Input:
a) $RiskScore$
b) $Frequency$
Output: LifeSpan of an IP.
procedure
    R=$\frac{Risk}{100}$
    F=$\frac{Frequency}{100}$
    if $R > 0.6$ && $F > 0.4$ then
        LifeSpan=90$^{+}$ days
    endif
    else
    if $R > 0.6$ && $F > 0.4$ then
        LifeSpan=30 days
    endif
    if $R < 0.6$ && $F > 0.4$ then
        LifeSpan=90 days
    endif
    return LifeSpan
```

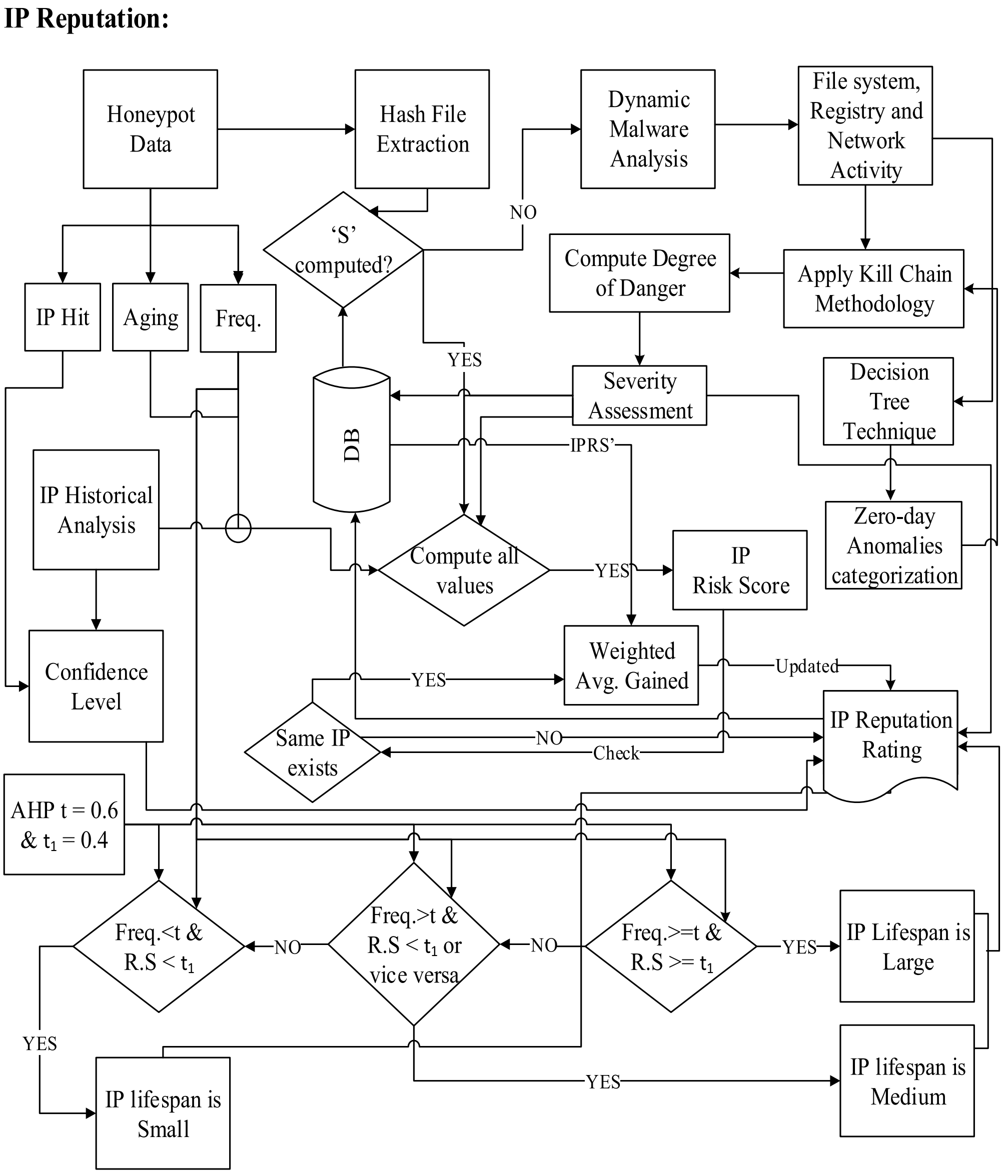

Fig. 3. Detailed flowchat of IP reputation system.

4. Performance evaluation

In this section, we present the results of the IP reputation framework. A list of binaries is executed in the cuckoo sandbox which returns the report against each binary. On these reports, we have applied the number of ML techniques for further analysis. A comparative study comprising Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Mini Batch K Means (MBK) has been presented. For the evaluations, we used a machine having the following specifications; 8 GB RAM, 2.3 GHz Processor, 500 GB HardDisk, Java 1.8, and Intellij IDE.

Binaries of 8 months were collected from September 2019 to April 2020 for both DB1 and DB2 and executed in cuckoo to achieve the overall behavior of malicious file. The binaries executed in the sandbox return the behavior, which depicts the file system, network-based, and registry-based of a specific delivered file. We have constructed two datasets; one contains the information regarding network activity for a list of malware files, whereas; the other database contains not only network-related information but also considers file system and registry-based activities. The reason behind the formation of two datasets is to evaluate network-based information against another dataset that contains the elaborated information. To attain more knowledge regarding big data analytics, we have considered both of the aforementioned datasets. The aforementioned ML techniques are applied to both of the datasets. First, we evaluate ML techniques on $Dataset_1$ that consist of limited information such as network-based activity. For evaluating ML techniques, we have considered several evaluative measures listed below:

i. Precision: It is also known as positive predictive value. A ratio of relevant rows to the retrieved rows in a file [62].

ii. Recall: It is also known as sensitivity. A ratio of relevant rows to the total relevant rows in the file [62].

iii. F-measure: It is also known as F-score. It is the harmonic mean of precision and recall [62].

Table 4
Performance comparison table using *Dataset*$_1$.

	F-measure	Precision	Recall
NB	75%	96%	62%
SVM	91%	98%	85%
DT	78%	75%	80%
MBK	75%	73%	78%

4.1. Experimental setup

As a virtual machine, the honeypot is deployed whose job is to log as much information as it can about the attacks. It is a trap for attackers to attract hackers and examine their behaviors. With the help of Docker, honeypots are organized. Docker is an open-source platform used for making, executing, and issuing applications that are introduced on a Virtual Private Server (VPS). The VPS is provided by COMSATS University of Information Technology (CIIT) and was constituted on a subnet located outdoor CIIt's central firewall. The firewall was configured to govern the streaming to and from the VPS.

Honeypots imitate certain services (for instance, FTP, SSH, HTTPS, telnet, and so on). It logs and accepts all entering networks. For 30 days, the actions on honeypots were examined. To catch suspicious IP addresses, all streaming to the Internet was permitted without any obstruction. An IP address with the associated binary file is extracted as it is detected on the honeypot. The binary file is the malicious file consisting of some suspicious activity whose aim is to harm the victim's machine or network. The detected binary file becomes an input for the phase of malware analysis. A binary file is executed in a secure environment, where the debugger listens to the traffic produced by the malware. Every action is captured by the debugger. The traffic includes behavior or activities performed by the malware. Activities can be related to network, file system, and registry-based actions. For the execution of a binary file, we prefer a Cuckoo sandbox that can detain every suspicious activity.

For gaining knowledge regarding IP addresses we dynamically analyze the malwares samples which are associated with the detected IP addresses, a sandbox named Cuckoo is used to trace the behavior of binaries. A list of binaries with MD5 is passed to the cuckoo and executed there. An entire activity of the binary file is captured by the cuckoo sandbox. Information regarding files, processes, registry, and the network is logged and retained in a file. Desired features are extracted from the output file to avoid redundancy and overhead.

From the extracted results we get some labeled and unlabeled binaries. The report having a name for malware depicts the malwares which are known to the antivirus software, while the one whose name is unknown is considered as zero-day malware. These unknown binaries are not labeled by antivirus software as they are new and there is no signature defined for them.

4.2. Experimental results on network dataset

In the Network dataset we have considered the information only regarding network activity of malicious binaries. The purpose of considering networks data is to compare the results of network dataset with the other dataset to follow the optimum approach in the proposed IP Rep. - FDA framework. The above-listed measures are considered for examining the performance of ML techniques on network data (*Dataset*$_1$) as shown in Table 4.

From Table 4, it can be seen that classification results are far better than the clustering results. Among the listed techniques, SVM performs better than the other techniques. The best error rate with maximum accuracy is produced by the SVM however; it consumes more time and requires more storage space than NB and DT. NB prediction rate is very high due to its biased nature which makes a bad impression for the NB algorithm. MBK is a clustering technique that not only consumes maximum time but also lags in predicting the true positive value. It is very difficult to set 'k' number of clusters for accurate partitioning. MKB distributes the dataset in batches due to which partitions produced at the end diverge from the true partition which indicates that MBK does not result in an optimal solution. According to Table 4, the false-negative rate for SVM is 15%, DT is 20%, MBK is 22%, and NB is 38%. SVM introduces a reduced false positive rate while comparing with other ML techniques.

Table 5
Performance comparison table using *Dataset*$_2$.

	F-measure	Precision	Recall
NB	72.5%	75%	70%
SVM	90%	98%	84%
DT	99%	100%	98%
MBK	78%	77%	84%

Table 6
Correctly identified versus incorrectly identified classes in *Dataset*$_1$.

	Correct prediction	Incorrect prediction
NB	64%	36%
SVM	82%	18%
DT	93%	07%
MBK	65%	35%

4.3. Experimental results on extended dataset

In the extended dataset we have considered the extended level of information regarding malicious binaries. The purpose of considering extended dataset is to compare the results of extended dataset with the network dataset to evaluate the optimum approach in the proposed IP Rep. - FDA framework. *Dataset*$_2$ contains not only network-based information but also considers file system and registry-based information. We have made a comparison among ML techniques by considering the extended dataset (*Dataset*$_2$) as shown in Table 5.

From Table 5 results, we can claim that as the number of features increases the performance of algorithms also varies with it. A dataset with increased number of features shows that the DT performs explicitly better than other techniques. SVM performance decreases with the increment in the selected attribute. As the number of selected features increases, SVM demands more time, and more storage space. By comparing Tables 4 and 5, we have discovered that the True positive rate for all selected techniques increases with the increment in the dataset. Prediction rate for the DT and MBK enhances which means that as many essential attributes we select the more accurate we get the result. However, setting 'k' in MBK is still a tough task whether the dataset is in KBs or MBs. Here, DT has the highest recall and precision rate which indicates that the DT has a lower error rate. According to Table 5, the false negative rate for DT is 2%, SVM and MBK is 16%, and NB is 30%. By comparing these ML techniques, we have seen that DT reduces false positive rate.

4.4. Experimental results of correct identification versus incorrect identification for network dataset

In this section, we examine classes which are correctly and incorrectly predicted by ML techniques for *Dataset*$_1$. Table 6 is illustrated to present the prediction rate.

Table 7
Correctly identified versus incorrectly identified classes in *Dataset*$_2$.

	Correct prediction	Incorrect prediction
NB	73.4%	26.6%
SVM	81.4%	18.6%
DT	93.5%	6.5%
MBK	75%	25%

Table 8
Performance in DT on network dataset i.e., *Dataset*$_1$ and extended dataset i.e., *Dataset*$_2$.

	F-measure	Precision	Recall
DT (network dataset)	78%	75%	80%
DT (extended dataset)	99%	100%	98%

Table 6 shows the comparison between correctly and incorrectly classified classes identified by ML techniques for limited selected features. DT presents an extremely better prediction rate than other classification and clustering techniques. After DT, SVM accurately predicts the classes with less error rate but consumes more time. NB and MBK are close to each other and produce a higher error rate than SVM and DT.

4.5. Experimental results of correct identification versus incorrect identification for extended dataset

In this section, we examine classes that are correctly and incorrectly predicted by ML techniques for *Dataset*$_2$. Table 7 is illustrated to present the prediction rate.

Table 7 shows us the comparison between correctly and incorrectly classified classes identified by ML techniques for extended features. The performance of DT is enhanced than DT's performance in Table 6. From this we can conclude that if the number of essential features increases then the DT's accuracy also improves with it. Likewise, MBK's and NB's prediction rate also improves with the increment in the selection of features. DT's, MBK's and NB's performance is directly proportional to the number of essential selected features. SVM's performance little bit decreases than DT's performance and demands more resources as the number of features increases. It means that SVM's performance is also directly proportional to the number of essential selected features.

$$DT_p \propto E.S.F$$

$$MBK_p \propto E.S.F$$

$$NB_p \propto E.S.F$$

$$SVM_p \propto E.S.F$$

where p indicates performance and *(E.S.F)* stands for Essential Selected Features. This is how; we are up to the statement that as we have more essential features we will achieve more accurate results. The more accurate results and optimum solution we get from DT, so the next step is to compare DT on both datasets including *Dataset*$_1$ and *Dataset*$_2$.

4.6. Network dataset i.e., *Dataset*$_1$ versus extended dataset i.e., *Dataset*$_2$

To evaluate the performance of DT algorithm on network and extended dataset we have applied DT on both datasets. The resultant positive predictive value, sensitivity and F-score are presented in Table 8.

Table 8 presents the comparative results for both network and extended datasets. As we can see that DT prediction rate is high when we use the extended dataset as an input for the DT algorithm. Furthermore, sensitivity scores also improved. The error rate produced by the DT on extended data is less as compared to the error rate produced by the DT on network dataset.

4.7. Extracting information from internal sources using extended dataset

In order to compute reputation score for plurality of IP addresses, we need to extract useful information from internal sources. For getting knowledge about an IP address from internal sources, we need to assess certain factors such as; severity, frequency and aging score. The aforementioned factors are briefly discussed below:

4.8. Severity assessment using extended dataset

After evaluating ML techniques, the next step is to assign severity to every instance. An instance basically represents the behavior of specific malware file. The extended dataset is used as an input for the assessment of severity score. Severity score is assigned on the basis of Kill Chain Methodology as discussed in Section 3.

4.9. Frequency and aging for the plurality of IP address using extended dataset

After assigning severity score to the behavior of malware instances, the next task is to compute the frequency and aging of associated IP address for computing the reputation of an IP address. The graph for an IP address 172.20.16.14 is presented in Fig. 4.

Fig. 4 shows the behavior of an IP address from 9/9/2019 to 1/7/2020. The severity, frequency and aging are the factors for which the information can be extracted from internal sources. There are some factors such as risk score, confidence level and lifespan for which external sources are required to be cross referred. In order to completely repute IP addresses we need to perform experiments on every considered factor. Therefore, the next step is to compute the risk score of an IP address.

4.10. Experimental results of risk score

In order to proceed our study, the next step is to assess the risk level for the plurality of IP addresses. For computing the risk level we provide internal and external sources as an input to the system. We are considering both internal and external sources to generate more accurate results. No organization can totally rely on the external parties (for instance; Virustotal, OTX, MyIP) information as these parties may call an IP address normal but that IP address may be discovered with malicious activity due to which our study claims that IP address should be considered suspicious IP.

An IP address associated with different kinds of malwares can attack multiple times on a system. The IP address that attacks again and again is more risky than the one that whose frequency is low. In order to demonstrate the risk level of an IP address that attempts multiple times, we have considered a case where we assign weights to the malicious IP address.

4.11. Case I: Weighted risk score

Risk score for every IP address is computed and stored in the database. A case occurs where a risk score is already calculated

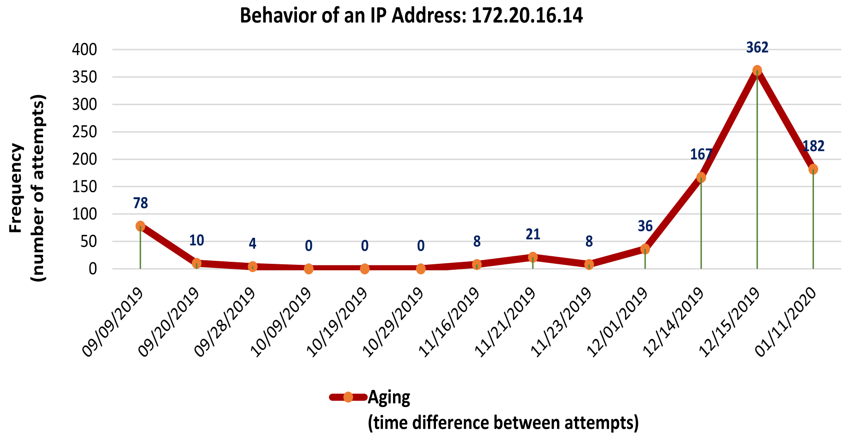

Fig. 4. Behavior of an IP address 172.20.16.14.

for the detected IP address then a mechanism is designed that assigns weights to the previously calculated risk score and recently calculated risk score of an IP address. After assigning weights a new risk score is assigned to the IP address as shown in Fig. 5.

As we can see in Fig. 5 that risk score for an IP address 172.20.16.14 is updated. Previously its risk score was computed but due to its re-attempt at a certain time we assign weights to the previously calculated risk score and newly calculated risk score to maintain the updated risk score. This is how; we achieve a weighted risk score. Rather than appending an IP's risk score again and again in the database we update its risk value to avoid the demand of more storage space. By doing so efficiency is achieved as we utilize the same memory addresses again and again on the reactivation of specific IP addresses. The location where an IP address was previously stored with its record is updated at the same location for further attempts.

To claim the maliciousness of an IP address with more confidence our study aims to compute the confidence score by using internal and external sources. To proceed our study, the experimental results for the confidence score are discussed in the next section.

4.12. Experimental results of confidence score

Further, we compute Confidence Level to assure the suspiciousness of an IP address. Rather than relying on a single source we use external sources to improve the guarantee level for a claim that states "an IP address is malicious". The experimental results of confidence score for the plurality of IP addresses are shown in Fig. 6.

Fig. 6 illustrates the confidence level for multiple IP addresses. The more it is listed in external parties the more we are confident about the malicious level of an IP address. As shown IP address 213.186.33.3 has a confidence level 100 which is greater than the confidence level of IP address 61.142.173.74 which is 25.

Like a weighted risk score, our study aims to compute the updated confidence level as well to assess the maliciousness of an IP address in external sources. In order to demonstrate the confidence level of an IP address that attempts multiple times, we have considered a case where we recompute the confidence level of malicious IP address

4.13. Case I: Updated confidence level

A case occurs where an IP address re-attempts to attack; its confidence level is computed again to test its behavior in external sources. As we claim that external parties information is not updated at run time and updated after some specific time interval thus any organization cannot totally rely on these sources. In order to portray the aforementioned limitation a graph is shown in Fig. 7.

Fig. 7 indicates that IP addresses were not detected as malicious in the previous report but according to new reports these IP addresses are detected malignant. Thus, these external sources lead towards false negative rate. To reduce the false negative rate we assign a weight to every source of information. On the re-attempt of an IP address we cross refer an IP's record again in the external sources information to be more confident regarding suspicious activity of an IP address. The confidence level of a specific IP address will remain the same if it does not appear again. On every re-occurrence of a particular IP address the status of IP address is cross checked in the external sources to again compute the confidence level. By doing so, more confidence is built on the suspicious activity of an IP address.

After the computation of confidence level, the proposed study aims to address lifespan of an IP address. To predict the life of an IP address, we have conducted experiments which are shown in the next section.

4.14. Experimental results of IP lifespan

Finally, the results for IP's lifespan need to be addressed. For predicting a TTL score for an IP address; risk and number of occurrences (frequency) of an IP address needs to be computed. As discussed in Section 3, the TTL is of three types; short, medium and long TTL. The TTL for list of IP addresses is shown in Fig. 8.

Fig. 8 illustrates the TTL for the number of IP addresses. There are 3 cases where TTL varies as shown below:

Case I: The TTL for IP address 192.168.x.y is long which means more than 90 days. The particular IP address can be seen within and after 90 days as its risk and frequency scores are high which alternatively means that particular IP address is very active in accomplishing its malicious activity.

Case II: The TTL for IP address 192.168.x.y is medium which means that within 90 days this IP can re-attempt to attack. The particular IP address can be seen within 90 days as its risk score is high and frequency score is low or may a risk score is low and frequency score is high.

Case III: The TTL for IP address 192.168.x.y is short which means till 30 days this specific IP can activate itself. The particular

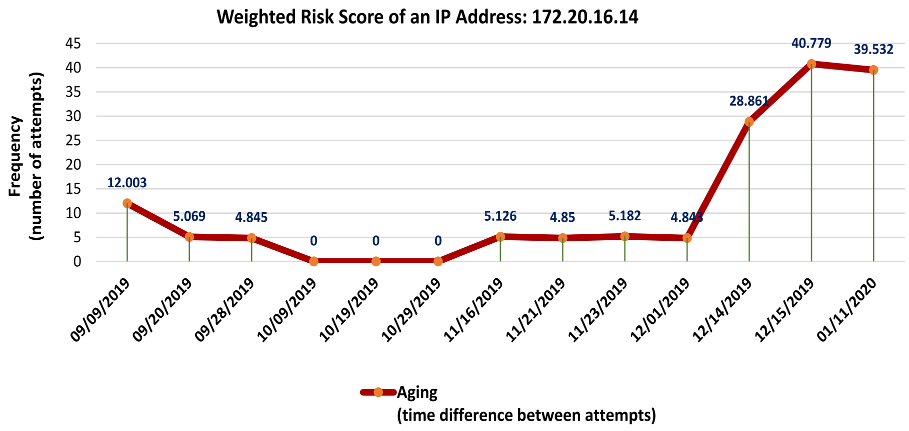

Fig. 5. Weighted risk score an IP address 172.20.16.14 w.r.t time.

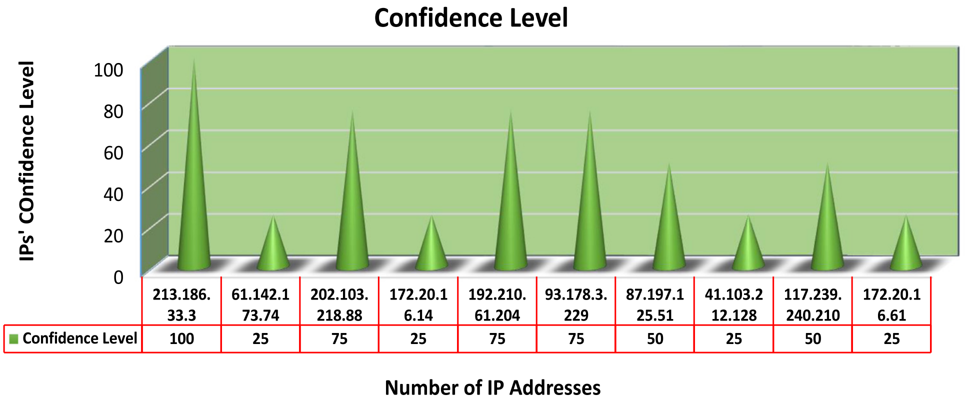

Fig. 6. Confidence level for the sample set of 10 IP addresses.

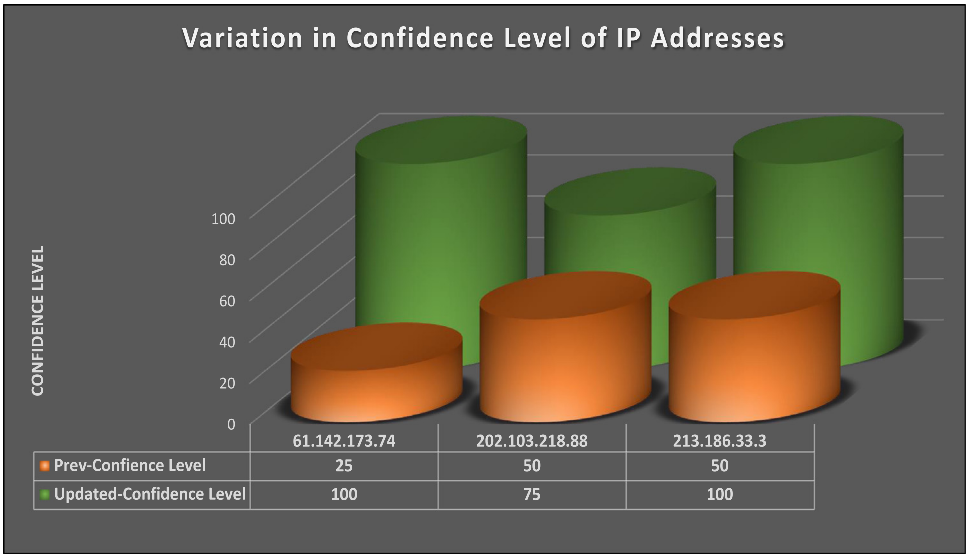

Fig. 7. Variation in confidence level w.r.t time.

IP address can be seen within 30 days as its risk and frequency scores are low which alternatively means that particular IP address is not malicious till that extent to be fitted in case I or case II.

After analyzing our main phases, we need to evaluate our IP reputation framework by comparing it with existing reputation frameworks. In order to test the main difference between the features provided by our reputation system and existing reputation systems we conduct a comparison as shown in Table 9.

From Table 9, we can conclude that some of the existing reputation systems do not employ external sources due to which they are not preferred as they have limited information regarding

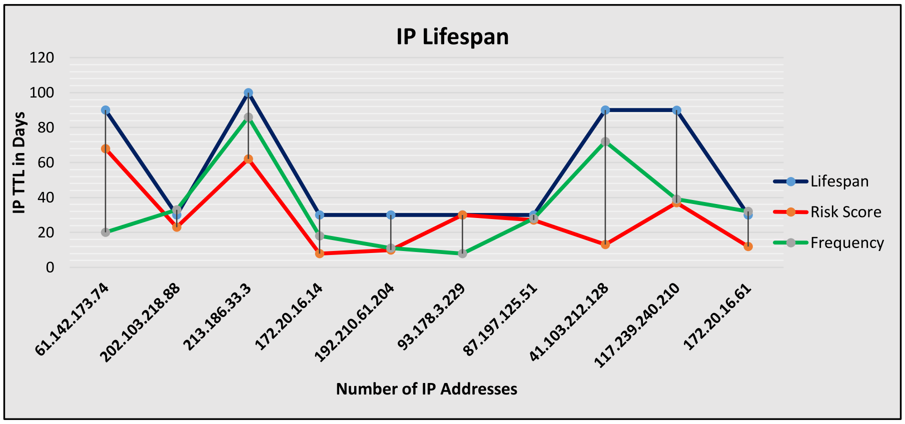

Fig. 8. Predicting TTL for the sample set of 10 IP addresses.

Table 9
Performance comparison of proposed IP Reputation for Forensics Data Analytics (IP Rep. - FDA) with existing reputation frameworks.

	Internal sources	External sources	Runtime Exec.	Severity score	Risk score	Confidence level	Lifespan	Ref.
Cyren	✓	✗	✓	✓	✓	✗	✗	[63]
Neustar	✓	✓	✓	✗	✓	✗	✗	[64]
Cisco Talos	✓	✗	✓	✓	✓	✗	✗	[65]
Barracuda	✓	✓	✓	✓	✓	✗	✗	[66]
Reputation authority	✓	✓	✓	✗	✓	✗	✗	[67]
IP Rep. - FDA	✓	✓	✓	✓	✓	✓	✓	–

any IP's risk score. In order to reduce the false negative rate, it is better to compute the reputation score for an IP address at run time. For instance, if a retroactive approach is followed for computing the reputation score of an IP address then this may lead towards the disastrous result. The reason is that an IP address may cause maximum damage and is shown still white listed in the databases of reputation engines. Thus, it is very essential for the reputation engines to be updated continuously. Likewise, some of the existing systems do not cater severity scores for malware samples as they may be associated with the plurality of IP addresses which should be addressed while computing the severity score. Furthermore, we can see that the proposed IP reputation framework is able to provide two more features which are not considered by any other reputation system. By having these two features/parameters, we have not only built a trust level for our system but also forecast a TTL for the spotted IP address which further improves the assurance level.

By comparing the proposed IP Rep. - FDA with existing reputation frameworks we have found that our framework produces far better results than other approaches. By deploying IP Rep. - FDA, any organization can efficiently detect the malicious IP addresses with more confidence. The existing frameworks depends only upon internal sources and once the IP address is considered suspicious its risk score is computed. The IP address may become more risky with time but the risk level is not updated by these reputation engines frequently which may lead towards high false alarm rate. Secondly, these frameworks do not predict the TTL of any IP address to show its recurrence lifespan. Whereas, our proposed IP Rep. – FDA considers both internal and external sources to reflect the results with more confidence. It computes the severity of a malware associated with the IP address which contributes to the computation of risk score. It also updates the risk score, confidence level and lifespan of an IP address for each of its suspicious attempts. In order to protect any security concerned organization it is better to deploy IP Rep. – FDA because it is able to generate the most updated results regarding suspicious IP addresses.

5. Conclusion

Cyber security is of main concern nowadays as everything is revolving around the Internet of Things. To interact in a network, a device must be well reputed by having a white listed IP address. In order to repute an object (such as IP address, DNS, User, Files, etc.,) number of reputation engines, blacklist IPs, domains, Indicators of Compromise (IOC), and so forth, are provided. However, these systems and approaches still lag behind in reputing an object due to several reasons (such as blacklists are outdated list due to management cost, reputation systems rely on only internal sources and returns high false negative rate, false reputation claims provided on small data, uses inappropriate classifier, following retroactive approach and so on). To repute an IP address at runtime, we have considered the big data and dynamically executed its associated binary file in cuckoo sandbox. Several ML techniques such DT, SVM, MBK and NB are applied on the dataset obtained from cuckoo. The aforementioned ML techniques are evaluated on the dataset and we have discovered that DT performs best while categorizing the unknown malware samples. After categorization, a kill chain methodology is applied to assign severity to a malware sample. In order to deal with big data, internal and external sources are considered for computing the risk score of an IP address. We have also computed confidence level and lifespan for assuring users to have a confidence on the IP reputation framework and protect them from the cyber threats.

Future work

In the study conducted above the false alarm rate is reduced up to some extent however, it is not diminished completely. In future, we desire to diminish the false alarm rate completely. Moreover, we aim to design rules from the discovered patterns obtained after applying the DT algorithm. The rules will trigger an alarm as the framework detects any bad reputed IP address. Detection of bad reputed IP addresses will enrich the protection of the system from any kind of suspicious activity. By doing so, we will be able to provide the enhanced policies which can be easily embedded in the anti virus or firewalls engines.

CRediT authorship contribution statement

Nighat Usman: Methodology, Software, Original draft preparation. **Saeeda Usman:** Conceptualization, Methodology, Validation. **Fazlullah Khan:** Writing - original draft, Validation, Formal analysis, Writing - review & editing. **Mian Ahmad Jan:** Data curation, Methodology, Writing - original draft. **Ahthasham Sajid:** Conceptualization, Methodology, Editing. **Mamoun Alazab:** Data curation, Methodology, Editing. **Paul Watters:** Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Alam, B. Rababah, Convergence of MANET in communication among smart devices in IoT, Int. J. Wirel. Microw. Technol. (IJWMT). 9 (2) (2019) 1–13, http://dx.doi.org/10.5815/ijwmt.2019.02.01.

[2] M.A. Khan, K. Salah, IoT Security: Review, blockchain solutions, and open challenges, Future Gener. Comput. Syst. 82 (2018) 395–411.

[3] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2671–2701.

[4] M.D. Harris, K.D. Ray, assignee inventors; Sophos Ltd, Threat detection using reputation data, 2017, United States patent US 9, 740, 859.

[5] R. Sihwail, K. Omar, K.Z. Ariffin, A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis, Int. J. Adv. Sci. Eng. Inf. Technol. 8 (4–2) (2018) 1662.

[6] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, C. Talhi, An anomaly detection system based on variable N-gram features and one-class SVM, Inf. Softw. Technol. 91 (2017) 186–197.

[7] R. Wason, A. Soni, M.Q. Rafiq, Estimating software reliability by monitoring software execution through opcode, Int. J. Inf. Technol. Comput. Sci. (IJITCS) 7 (9) (2015) 23–30.

[8] S. Arshad, M.A. Shah, A. Khan, M. Ahmed, Android malware detection & protection: a survey, Int. J. Adv. Comput. Sci. Appl. 7 (2) (2016) 463–475.

[9] V. Bartos, M. Zadnik, S.M. Habib, E. Vasilomanolakis, Network entity characterization and attack prediction, Future Gener. Comput. Syst. 97 (2019) 674–686.

[10] T. Wuchner, M. Ochoa, M. Golagha, G. Srivastava, T. Schreck, A. Pretschner, Malfow: identification of c & c servers through host-based data ow profiling, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, ACM, 2016, pp. 2087–2094.

[11] R.P. Jover, I. Murynets, assignee inventors; AT & T Intellectual Property I LP, Malware and anomaly detection via activity recognition based on sensor data, 2019, United States patent US 10, 516, 686.

[12] A.H. Johar, A. Gerard, N. Athar, U. Asgher, Feature based comparative analysis of online malware scanners (OMS), in: International Conference on Applied Human Factors and Ergonomics, Springer, Cham, 2020, pp. 385–392.

[13] D. Sun, Z. Wu, Y. Wang, Q. Lv, B. Hu, Risk prediction for imbalanced data in cyber security: A siamese network-based deep learning classification framework, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[14] O. Yavanoglu, M. Aydos, A review on cyber security datasets for machine learning algorithms, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2186–2193.

[15] P.V. Amoli, T. Hamalainen, G. David, M. Zolotukhin, M. Mirzamohammad, Unsupervised network intrusion detection systems for zero-day fast-spreading attacks and botnets, Int. J. Digit. Content Technol. Appl. 10 (2) (2016) 1–13.

[16] A.A. Nasr, M.M. Ezz, M.Z. Abdul maged, An intrusion detection and prevention system based on automatic learning of traffic anomalies, Int. J. Comput. Netw. Inf. Secur. 8 (1) (2016) 53.

[17] Y. Zhang, X. Wang, H. Xie, W. Xu, assignee inventors; Palo Alto Networks Inc, Malware domain detection using passive DNS, 2019, United States patent US 10, 237, 283.

[18] K. Alieyan, A. ALmomani, A. Manasrah, M.M. Kadhum, A survey of botnet detection based on DNS, Neural Comput. Appl. 28 (7) (2017) 1541–1558.

[19] Guarnieri C. Cuckoo, http://www.cuckoosandbox.org/ (Accessed 19-June-2019).

[20] R. Kaur, M. Singh, Hybrid real-time zero-day malware analysis and reporting system, J. Inf. Technol. Comput. Sci. (IJITCS) 8 (4) (2016).

[21] R. Smith, S. Marck, assignee inventors; Level 3 Communications LLC, Identifying a potential DDOS attack using statistical analysis, 2018, United States patent US 9, 900, 344.

[22] B. Yanovsky, S. Eikenberry, assignee inventors; SonicWALL Inc, Reputation-based threat protection, 2019, United States patent US 10, 326, 779.

[23] T. Chin, K. Xiong, C. Hu, Y. Li, A machine learning framework for studying domain generation algorithm (DGA)-based malware, in: International Conference on Security and Privacy in Communication Systems, Springer, Cham, 2018, pp. 433–448.

[24] L. Wang, B. Wang, J. Liu, Q. Miao, J. Zhang, Cuckoo-based malware dynamic analysis, Int. J. Perform. Eng. 15 (3) (2019).

[25] D.L. Hillard, A. Munson, L. Cayton, S. Golder, inventors eSentire Inc assignee, System and method for determining network security threats, 2019, United States patent US 10, 412, 111.

[26] K.D. Ray, S.N. Reed, M.D. Harris, N.R. Watkiss, A.J. Thomas, R.W. Cook, D. Samosseiko, assignee inventors; Sophos Ltd, Using indications of compromise for reputation based network security, 2018, United States patent US 9, 992, 228.

[27] Y. Li, F. Liu, Z. Du, D. Zhang, A simhash-based integrative features extraction algorithm for malware detection, Algorithms 11 (8) (2018) 124.

[28] V.V. Thang, F.F. Pashchenko, Multistage system-based machine learning techniques for intrusion detection in wifi network, J. Comput. Netw. Commun. (2019).

[29] R. Vinayakumar, P. Poornachandran, K.P. Soman, Scalable framework for cyber threat situational awareness based on domain name systems data analysis, in: Big Data in Engineering Applications, Springer, Singapore, 2018, pp. 113–142.

[30] C. Kruegel, L. Bilge, E. Kirda, M. Balduzzi, Exposure: finding malicious domains using passive DNS analysis, in: Proc. of 18th Network and Distributed System Security Symp. NDSS'11 2019 (pp. 214-231).

[31] J.U. Joo, I. Shin, M. Kim, Efficient methods to trigger adversarial behaviors from malware during virtual execution in sandbox, Int. J. Secur. Appl. 9 (1) (2015) 369–376.

[32] R. Sharifnya, M. Abadi, A novel reputation system to detect dga-based botnets, in: Computer and Knowledge Engineering (ICCKE), 2013 3th International EConference, IEEE, 2013, pp. 417–423.

[33] B. Coskun, (Un) wisdom of crowds: Accurately spotting malicious IP clusters using not-so-accurate IP blacklists, IEEE Trans. Inf. Forensics Secur. 12 (6) (2017) 1406–1417.

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

[35] Y. Nakamura, S. Kanazawa, H. Inamura, O. Takahashi, Classification of unknown web sites based on yearly changes of distribution information of malicious IP addresses, in: 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2018, pp. 1–4.

[36] A. Niakanlahiji, M.M. Pritom, B.T. Chu, E. Al-Shaer, Predicting Zero-day Malicious IP Addresses. in: Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense 2017 Nov 3 (pp. 1-6).

[37] J.L. Lewis, G.F. Tambaliuc, H.S. Narman, W.S. Yoo, Reputation analysis of public databases and machine learning techniques, in: 2020 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2020, pp. 181–186.

[38] N. Kaur, A.K. Bindal, A complete dynamic malware analysis, Int. J. Comput. Appl. 135 (4) (2016) 20–25.

[39] D. Ucci, L. Aniello, R. Baldoni, Survey of machine learning techniques for malware analysis, Comput. Secur. 81 (2019) 123–147.

[40] B. Kolosnjaji, A. Zarras, G. Webster, C. Eckert, Deep learning for classification of malware system call sequences, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2016, pp. 137–149.

[41] N. Koroniotis, N. Moustafa, E. Sitnikova, Forensics and deep learning mechanisms for botnets in internet of things: A survey of challenges and solutions, IEEE Access 7 (2019) 61764–61785.

[42] J. Zhou, Z. Cao, X. Dong, A.V. Vasilakos, Security and privacy for cloud-based IoT: Challenges, IEEE Commun. Mag. 55 (1) (2017) 26–33.

[43] D. Zhan, L. Ye, B. Fang, H. Zhang, X. Du, Checking virtual machine kernel control-flow integrity using a page-level dynamic tracing approach, Soft Comput. 22 (23) (2018) 7977–7987.

[44] D. Vidyarthi, S.P. Choudhary, S. Rakshit, C.S. Kumar, Malware detection by static checking and dynamic analysis of executables, Int. J. Inf. Secur. Priv. (IJISP) 11 (3) (2017) 29–41.

[45] O. Or-Meir, N. Nissim, Y. Elovici, L. Rokach, Dynamic malware analysis in the modern era—A state of the art survey, ACM Comput. Surv. 52 (5) (2019) 1–48.

[46] B. Sun, A. Fujino, T. Mori, T. Ban, T. Takahashi, D. Inoue, Automatically generating malware analysis reports using sandbox logs, IEICE Trans. Inf. Syst. 101 (11) (2018) 2622–2632.

[47] S. Kaur, M. Singh, Hybrid intrusion detection and signature generation using deep recurrent neural networks, Neural Comput. Appl. 11 (2019) 1–9.

[48] M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, Z. Chen, Efficient signature generation for classifying cross-architecture IoT malware, in: 2018 IEEE Conference on Communications and Network Security (CNS), IEEE, 2018, pp. 1–9.

[49] R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system, IEEE Access 7 (2019) 41525–41550.

[50] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (2019) 20.

[51] Z. Sun, Z. Rao, J. Chen, R. Xu, D. He, H. Yang, J. Liu, An Opcode sequences analysis method for unknown malware detection, in: Proceedings of the 2019 2nd International Conference on Geo Informatics and Data Analysis 2019 Mar 15 (pp. 15-19).

[52] S.M. Bidoki, S. Jalili, A. Tajoddin, Pbmmd: A novel policy based multi-process malware detection, Eng. Appl. Artif. Intell. 60 (2017), 57–70.4.

[53] B. Cakir, E. Dogdu, Malware classification using deep learning methods, in: Proceedings of the ACMSE 2018 Conference 2018 Mar 29 (pp. 1-5).

[54] M. Hafsa, F. Jemili, Comparative study between big data analysis techniques in intrusion detection, Big Data Cogn. Comput. 3 (1) (2019) 1.

[55] S.Y. Choi, C.G. Lim, Y.M. Kim, Automated link tracing for classification of malicious websites in malware distribution networks, J. Inf. Process. Syst. 15 (1) (2019).

[56] Virustotal, Virustotal, 2020, https://www.virustotal.com/ (Accessed 27-June-2020).

[57] OTX, Alienvault, 2020, https://www.alienvault.com/openthreat-exchange (Accessed 27-June-2020).

[58] MyIP.MS, 2020, https://www.myip.ms/browse/blacklist/ (Accessed 27-June-2020).

[59] T. Mithal, K. Shah, D.K. Singh, Case studies on intelligent approaches for static malware analysis, in: Emerging Research in Computing, Information, Communication and Applications, Springer, 2016, pp. 555–567.

[60] D. Kiwia, A. Dehghantanha, K.K. Choo, J. Slaughter, A cyber kill chain based taxonomy of banking trojans for evolutionary computational intelligence, J. Comput. Sci. 27 (2018) 394–409.

[61] F. Gong, Fengmin, D. Jas, MacFarlane, System and method for threat risk scoring of security threats, 2016, US Patent 8, 832, 832.

[62] A.A.A. Rostamy, D. Khosroanjom, A. Niknafs, A.A. Rostamy, Fuzzy AHP models for the evaluation of it capability, data quality, knowledge management systems implementation and data security dimensions, Int. J. Oper. Res. 22 (2) (2015) 194–215.

[63] L. Samuelson, CYREN Reputation, 2020, http://www.cyren.com/ip-reputation-check.html (Accessed 20-June-2020).

[64] P.D. Ballew, IP Reputation, 2020, https://www.neustar.biz/risk/compliance-solutions/ip-intelligence/ip-reputation (Accessed 20-June-2020).

[65] B. Shah, Cisco umbrella: A cloud-based secure internet gateway (SIG) on and off network, Int. J. Adv. Res. Comput. Sci. 8 (2) (2017).

[66] S.S. Ninawe, P. Venkataram, Authentication schemes for social network users: a review, Int. J. Soc. Comput. Cyber-Phys. Syst. 2 (2) (2019), 151–176.

[67] D. Preuveneers, W. Joosen, Managing distributed trust relationships for multi-modal authentication, J. Inf. Secur. Appl. 40 (2018) 258–270.

Nighat Usman is currently working as a Lecturer in the Department of Computer Sciences at Bahria University Lahore Campus. She did Masters in Information Security from CIIT Islamabad campus. She has gained a pretty good knowledge regarding IBM QRadar, OSSIM, Elasticsearch. Furthermore, she was rewarded with an Award by Trillium Pakistan for her research contribution in rule mining technique for profiling log behavior.

Saeeda Usman research interest is the designing and optimization of low power electronic devices. Her research interests encompass topics related to energy efficient scheduling in clusters, performance optimization, and robustness in Cloud Computing. She is currently working on the management and scheduling of resources in Cloud computing. Moreover, she is involved in the designing and analysis of the low power consuming computing chips and devices (for exploration of energy efficient solution to meet the escalating need of power/energy resources.

Fazlullah Khan is a researcher at Ton Duc Thang University Ho Chi Minh City, Vietnam, and working as an Assistant Professor of Computer Science at Abdul Wali Khan University Mardan, Pakistan. His research interests are Security & Privacy, Internet of Things, Machine Learning, Software-defined Networks, Fog Computing and Big Data Analytics. He has published his research work in top-notch journals and conferences. His research has been published in IEEE Transactions on Industrial Informatics, IEEE Transaction on Vehicular Technologies, IEEE Access, Elsevier Future Generations Computer Systems, Elsevier Journal of Network and Computer Applications, Elsevier Computers and Electrical Engineering, Springer Mobile Networks and Applications. He has served as the guest editor of IEEE Access Journal, Springer Multimedia Technology and Applications, Springer Mobile Networks and Applications.

Mian Ahmad Jan is an assistant professor at the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He has completed his Ph.D. in Computer Systems from University of Technology Sydney (UTS), Australia in 2016. He had been the recipient of various prestigious scholarship during his studies, notably the International Research Scholarship (IRS) at the UTS and Commonwealth Scientific Industrial Research Organization (CSIRO) scholarships. He has been awarded the best researcher awarded for the year 2014 at the UTS, Australia. His research interests include Security and Privacy in Internet of Things and Wireless Sensor Networks. His research has been published in various prestigious IEEE Transactions and Elsevier Journals. He has been the general Co-chair of Springer/EAI 2nd International Conference on Future Intelligent Vehicular Technologies, 2017. He has been guest editor of numerous special issues in various prestigious journals such as Elsevier Future Generation Computer Systems, Springer Mobile Networks and Applications (MONET), Ad Hoc & Sensor Wireless Networks, and MDPI Information.

Ahthasham Sajid is currently working as Assistant Professor in Department of Computer Science at Balochistan University of Information Technology Engineering and Management Sciences Quetta, PAKISTAN. His areas of interest are Computer Networks, Wireless & Sensor Networks, Mobile Communication, under water Sensor Networks and Network Security. He has 3 SCI indexed, 7 SCOPUS/ESCI indexed, 20 HEC Recognized Journal, 6 International and 2 National Conference Proceedings Publications.

Mamoun Alazab is an Associate Professor in the College of Engineering, IT and Environment, IT Discipline. He is a cybersecurity researcher and practitioner with industry and academic experience. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems including current and emerging issues in the cyber environment like cyber–physical systems and internet of things, by taking into consideration the unique challenges present in these environments, with a focus on cybercrime detection and prevention. Assoc. Prof. Alazab received his Ph.D. degree in Computer Science and has more than 100 research papers. He presented at many invited keynotes talks and panels, at conferences and venues nationally and internationally (22 events in 2018 alone). He is a Senior Member of the IEEE. He is an editor on multiple editorial boards including Associate Editor of IEEE Access (2017 Impact Factor 3.557), Editor of the Security and Communication Networks Journal (2017 Impact Factor: 0.904) and Book Review Section Editor: Journal of Digital Forensics, Security and Law (JDFSL).

Professor Paul Watters is Australia's leading cybersecurity researcher. He was recently awarded $2.364m from the MBIE Catalyst Strategic — Cyber Security Research by the New Zealand government to develop Artificial Intelligence for Automating Responses to Threats. This collaboration includes Data61, Victoria University of Wellington and the University of Queensland. He has also recently been awarded funding from the Oceania Cyber Security Centre (OCSC) for six projects: Automated Story Generation for Intelligible Cyber Incident Response Handling (with funding from Cisco) $253, Spam Email Categorisation Using Natural Language Processing and Attention-Embedded Deep Learning (with funding from Westpac) $160k, Adaptive API Security Using Artificial Intelligence (with funding from Aiculus) $139k, Real-time Zero Day Ransomware Attack Detection $79k, A Multi-Layered Approach to Detecting Malicious Mobile Advertising $78k, and Development of Smart Grid Cyber Security Testbed $69k.