2021 International Conference on Innovative Computing (ICIC)

A Systematic Literature Review: Usage of Logistic Regression for Malware Detection

Zeshan Akram
Department of Informatics and Technology
University of Management and Technology
Lahore, Pakistan
F2019108009@umt.edu.pk

Mamoona Majid
Department of Informatics and Technology
University of Management and Technology
Lahore, Pakistan
Mamoona.majid@umt.edu.pk

Shaista Habib
Department of Computer Science
University of Management and Technology
Lahore, Pakistan
Shaista.habib@umt.edu.pk

2021 International Conference on Innovative Computing (ICIC) | 978-1-6654-0091-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIC53490.2021.9693035

***Abstract*—Malwares are serious threats since decades and now they are becoming a huge risk due to the increasing nature of their attacks. At first computer virus named "brain" was introduced, which raised the need for security measurement. Later on, the malware and malicious content did not only breach the security measurements through attaching infected devices to computer systems but also approach via network usage. Nowadays malware is a more crucial and important topic that needs to be examined carefully to avoid security issues. Millions of new malwares are reported every year so we need a fast, reliable, and trustworthy solution against this malware. Machine learning techniques are very efficient and robust to recognize malicious malware attacks. Different malware detection approaches have been developing to overcome security issues. Among all, the Logistic Regression classifier is very suitable to deal with a large number of data sets available over the internet. This article provides a step-by-step approach to conducting a Systematic Literature Review (SLR) in the domain of malware detection. SLR assesses the question of interest-based on the quality level and magnitude of existing literature. Preferred reporting item for systematic reviews and meta-analysis is used here to create a framework for SLR and verify the quality of articles. All the papers collected from various resources such as IEEE Xplore, Wiley Library, ACM Microsoft, etc. will be able to detect malware attacks.**

Keywords—Malware, Machine Learning, Regression techniques, logistic regression, Support Vector Machine.

I. Introduction

This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document" for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. There is an endless conflict between users and malicious users but malware is always harmful to users. Therefore millions of Malwares have been spread in computer systems, networks, and the internet that can drive a user to an infected website or can launch Denial of Service Attacks (DDoS) to steal confidential and private data by using various techniques. Malware is harmful programs (Virus, Worm, Trojan horse, Rootkit, Backdoor, APT) that can cause damage to your computer [1-3]. Generally, malware has been analyzed statically and dynamically [4]. Static analysis is the process in which we examine the malware without execution. While the dynamic analysis is to examine the malware by using their actions and behavior after running a file or code onto a secure system. It may change the windows library and play with various organization exercises also. These viruses can perform several harmful activities like altering or stealing data, hijacking, and encryption, etc. [5-6].

Analysis techniques have evolved dramatically over the years, now they have much ability to detect the attacking mechanisms and methods. Therefore, authors invest much of their time to design new malware attacks of their own to avoid these analysis tools. We realize that malware is really a hot topic among malware authors. Moreover, the emerging behavior of malware not only misled the normal user but security administrators as well like a polymorphic, anti-sandbox, and file-less malware. Polymorphic are those that keep on repeating their internal line of code to avoid the fear of being captured. Anti- box is a technique in which a malicious program becomes able to detect itself during the analysis process but delays its execution until after it leaves the sandbox while file less hides into the primary storage of a computer to avoid being discovered [7]. New strategies and refined practices have evolved to overcome these problems [8].

In recent years, machine learning and data mining techniques have been used to provide different solutions to detect malware activities. Machine learning (ML) algorithms are used to detect malicious software or programs by predicting their behavior, generalization, and classification [9]. Moreover, it is very easy for the assailants to alter or expel a specific part of a computerized image with some other part without leaving any evidence using ML techniques. This malware can even destroy somebody's reputation and is also known as image imitation or vindictive malware [10]. Similarly, web applications are becoming the most standard platform for information representation. Web applications like banking, finance, healthcare, etc. are more vulnerable to the attack. In this area, Cross-site scripting (XSS) is the most vulnerable attack that allows the attacker to execute a malware-based script on the victim's web browser to steal personal information like passwords, credit cards, numbers, etc. [11].

AI can be utilized for bogus information ingestion to create imaginary information structures [12]. Banking sites are vulnerable to Trojan attacks and their detection is much more difficult due to constant evolutionary techniques used to avoid security solutions. Threat-based malware taxonomies have been designed specifically for banking Trojan attacks [13]. On the other hand, malware attacks on android applications are increasing day by day. To reduce their effect, multiple analyses have been done on the relevant

978-1-6654-0091-6/21/$31.00 ©2021 IEEE

features, extracted at the API level using ML techniques such as clustering algorithms, KNN classifiers, CNN, etc. [14-19].

In this paper, we will discuss different approaches of ML algorithms such as logistic Regression, and some other classifiers combined together for malware detection inside computer systems.

II. LITERATURE REVIW

Malware is a security threat to data that keeps on expanding. In 2014 almost 6,000,000 novel malwares were reported. Adware is malware that hides in advertisements and serves you on different platforms available. There are many types of Adware malware but Trojan horse is the most common of them. Often malware security is compromised due to various IoT devices but multiple security frameworks are available in the market to guard secure systems against these viruses. i.e., antivirus, firewalls, sandboxes, etc. Due to the huge proliferation of PCs, it is very difficult to distinguish between various types of malware. Intrusion detection systems also work to identify the different behavior of malware. Other than signature-based security frameworks it is hard to recognize new techniques, infections, or worms utilized by attackers [20].

Malware identification is the most crucial part of malware safety assurance. Some Data Mining techniques are available that play a significant role during the analysis of data. S. L Rao [21] examined malware location in secure systems utilizing logistic regression classifiers. The Logistic Regression (LR) classifier is endorsed to recognize obscure malware with the likelihood of 74 to 83 %. The identification criteria are that LR produces a recognition model from an adequate dataset of noxious software [21]. On the other hand, many malware problems are solved using Convolutional Neural Network (CNN). It has been observed that malware can be characterized with an 85% precision rate using AI techniques on 36 malware families having 12,279 samples. Additionally, Ozkan et al. have achieved 99 % accuracy on 25 families having 9,339 malware samples [22].

Cen et al. developed a probabilistic discriminative model for android malware detection using source code, permission, and dynamic analysis. First, they have extracted features using android API calls from decompiled source code and application permissions. After extraction, they developed a probabilistic model that outperformed with permission for android malware detection [23]. In [24], Kumar et al. developed a defensive system against polymorphic malware using ANOVA F- test technique. Instead of traditional signature- based malware, polymorphic malware is those which change their pattern after every attack. Using ANOVA F-test, they achieved 97.7% accuracy.

Suhuan et al. Suhuan et al. developed a system to detect malware in android based applications [25]. They developed a multi-dimensional feature-based application using logistic regression and the XGBoost method. First, they extracted probabilistic features by applying LR to train N-gram modeled API call sequences. After extraction, they combined them with statistical features as well and embedded them into XGBoost to detect malicious android malware. Malware attacks also play a significant role in the cyber-security domain. In this area, they often occur in the form of botnets. Botnet malware usually targets vulnerable devices connected with as many as possible devices rather than individual.

Bapat et al. proposed an anomaly-based intrusion detection system using statistical learning methods between source and destination [26]. Later, these statistical features were embedded into the LR model as an input to detect botnet activity within network traffic. Aside from shallow learners, deep learning methods also gave promising results in the Machine learning domain. Mausam et al. [27] developed a deep learning frame known as Droid-NNet for malware classification in android OS.

K. Kumar et al. developed a clustering-based malware detection system [28]. They used Cuckoo sandbox for dynamic analysis of files being executed on the system. After analysis, they extracted features using novel approaches like principal component analysis (PCA), chi-Square and random forest. After this, they trained clustering and non-clustering algorithms on three classifiers including random forest, decision tree, and logistic regression. The developed clustering model was outperforming the non-clustering model. Cam developed a malware detection system based on LR and a partially observable Markov decision process along with temporal and dependency relationships [29]. They have identified the cross relationships of malware activities and then measure the initial probability values from the sensor which infers the infection status of those assets who are likely to be exposed or vulnerable.

A hardware malware detector (HMD) is used to detect malware on a hardware level. These detectors usually use low-level monitoring features like CPU performance etc. Due to the proliferation of malware, it is very easy to avoid detection by HDMs. These detectors can be easily evaded or reverse-engineered. Khasawneh et al. showed that simple detectors such as logistic regression, cannot detect malware effectively and can be evaded easily [30]. They developed a novel detector known as resilient hardware malware detector (RHMD) that stochastically moves between different detectors to detect evasive malware. RHMD are very difficult to evade or reverse- engineered. They also used these special detectors to detect specific malware of each type [31]. Therefore, the increased accuracy of HMDs, to improve detection and reduce overheads using neural networks (NN). NN detectors outperform LR detectors by 40 % of accuracy. Tewari et al. developed a system for android malware detection using permission and API calls with an accuracy of 97.25% [32]. They extracted features on the basis of the common feature vector and combined vector.

In [33], authors developed a system to detect malware in android applications by classifying these applications on the basis of permission. They used six different ML algorithms to classify these applications into vulnerable, malicious, or benign applications. Logistic regression outperforms with an accuracy of 99.34 %. Similarly in [32], authors developed an android malware detection system using LR and Support Vector Machine (SVM). They achieved 97.72% accuracy having 350 features and 94.69% accuracy having 30 features using SVM. A combination of two or more ML algorithms has been used to improve the accuracy of malware detection in above approach [34].

Sasaki et al. developed an attack framework to prevent the detection of only a specific type of malware by data poisoning attack. They embedded backdoors into the system that generate miss-predictions [35]. Solving optimization problems, they improved 30% Detection Rate to avoid attack detection. Abaid et al. has performed statistical analysis of the effect of adversarial attacks on linear and non-linear classifiers [36]. They observed that attackers can easily reduce the detection rate from 100% to 0% by changing only a few features of malicious applications. They analyzed that nonlinear classifiers are more robust to attacks than linear classifiers with an accuracy of 99.1%.

In another paper, Samantray et al. developed a framework to detect malicious malware using ML techniques such as SVM, Logistic regression, and Naive Bayes [37]. They extracted the most relevant features for classification using K-best and extra tree classifiers. Logistic regression and Naive Bayes (NB) performed well and their model achieved 95.53% accuracy. Similarly, Wang et al. developed another framework "KerTDroid" to detect malware in parallel through kernel task structures in the Android kernel layer [38]. They used many ML algorithms like Decision tree (DT), NB, LR, and SVM. The decision tree method performed very well and achieved 96%-99% accuracy while the other three leads to a lower precision rate than 90%.

Ofori et al. developed a Cyber Supply Chain (CSC) system to detect malware-based nodes in the organization [39]. For this purpose, they used many ML algorithms (LR, DT, and SVM) to predict which nodes are more vulnerable to attacks or not. The decision tree performed very well to predict future cyber-attacks in CSC. Due to the frequent use of cloud infrastructure, malware attacks are increasing in this domain too. Win et al. performed security analyses on big data collected from Network and user application logs of VMs [40]. After extracting relevant attack features through the map-reduce parser, they performed logistic regression methods. Using logistic regression, they calculated the attack's probabilistic values to detect attacks in the virtualized environment with an accuracy of 98.84%.

Krishnan et al. has performed the same experiment over network log and user application logs of big data [41]. They achieved good performance on the average detection time rate. In [42], authors developed a platform to detect android based malware using different ML methods (DT, LR, K Nearest Neighbor, NB, RF, and SVM) as well as Deep learning methods like multi-layer perceptron. Performance is being measured using each algorithm, but RF and SVM provide the best results for malware detection with 91.69% accuracy. Like [40, 41] Gupta et al. have performed the same experiment on the data collected through VMs [43]. In [44] Amin et al. designed a model using a generative adversarial network (GAN) to detect malware in android devices. They extracted the dataset from the android package kit byte code and used GAN via two-player game theory for the rock-scissor-paper problem and achieved 99% accuracy.

Bae et al. [45] performed experimentation with ML algorithms and achieved accuracy of 90.27%. Kumar et al. designed an anti- malware solution for using system refined calls of android devices [46]. They proposed two novel approaches of feature selection called Rough set and statistical test (RSST). After extraction of relevant features, they performed different ML algorithms such as RF, NB, LR, and SVM and achieved an accuracy of 99.9%. Keong e al. developed an integrated framework where honeypot (security mechanism that creates virtual traps for attackers) is deployed in various frameworks like IPs gateways, analysis systems, etc. to detect ransom ware on the client side [47]. They achieved higher accuracy of 96.8% than other ML models.

In [48], authors experimented with deep learning neural network model (DNN) to detect seven different attacks found in Distributed systems via IoT network traffic. They achieved an accuracy of 97.01%. Xiong et al. developed a machine learning- based framework which used a domain generating algorithm (DGA) to identify and detect malware threats [49]. Their model consists of two components, first is used to classify DGA domains from normal domains. Then clustering methods are used to detect the algorithm that generates those DGA domains with an accuracy of 97.4%. Deng et al. designed a framework based on feature optimization and hybrid classification [50]. They extracted malicious web page features using gain based feature selection method. After extraction, they performed multiple integrated ML algorithms (achieved 95.75% accuracy) to detect malware on web pages.

Pascanu et al. proposed a unique technique that learns the behavior of malware through executed instructions in the system [51]. They extracted real-time features using echo state networks (ESN) and NNs. To classify malware, they used ESN with recurrent networks and Logistic regression. Performance is being measured using the TP rate of 98.3%. In another paper, the authors presented a detailed analysis of factors by observing five different families of malware [52]. They proposed a data-driven approach to categorize using both traditional and deep ML with an accuracy of 97%. In other paper [53], authors performed ontology approaches to detect problems in educational institutes. In work [54], authors have done a detailed survey on routing and forwarding techniques in distribution content network (DCN). Their survey reveals the problem in DCN using parameters like network protocols, topologies, traffic sensitive routing algorithms etc.

TRUE positive

After having an extensive literature survey, the Systematic literature review (SLR) method is described in next section. After SLR, the paper has been concluded efficiently.

III. Systematic Literature Reviw Methodology

① First, we formulated our research questions from reviewed articles ② then searched these keywords in various research databases. Using avoidance and consideration criteria, only relevant papers have been considered. ③ After this, we have assessed the quality of the papers and then we have extracted the relevant information.

*

A. Research Questions

We have designed some questionnaires related to the domain to find the quality research paper. Some of them are given below:

- RQ1: Which machine learning techniques are used for malware detection?
- RQ2: How efficient and successful are these techniques?
- RQ3: What are the limitation and future directions?

B. Research Databases

After RQs, we have collected research papers from different repositories like IEEE Explore, ACM, Wiley Online Search, Base- Search, Google Scholar, and Microsoft Academic, etc. using the SLR method mentioned earlier.

C. Inclusion and Exclusion Process

Thousands of research papers have been found related to the domain. To choose relevant and concise data we choose the following eligibility criteria.

1) Consideration Criteria: The papers which are publicly available after 2017 onwards, non- paid and recognized by reputed journals and conferences have been considered to pursue further.

2) AvoidanceCcriteria: Most of the research papers are obsolete, therefore we avoid those papers and the ones which are not publicly available or published before 2017. Table I shows databases and year-wise paper distribution of the considered articles.

D. Quality Assessment

In this phase, we have analyzed research papers on the basis of multiple criteria like their objective, datasets and methods used are clearly stated, feature selection etc. as shown in Table II.

E. Information extraction

After collection, some important and significant features have been analyzed and extracted e.g.; Year of distribution, AI methods, performance, accuracy and future work as shown in Table III.

F. Information Synthesis

The information or the features collected from the previous section integrate with responses collected through questionnaires. Information synthesis has been performed by examining the literary works through different measurements [53]. Table IV shows article- wise limitations and futures directions.

To describe SLR in detail, we have designed Table III which addresses RQ1 and RQ2 and Table IV addresses RQ3.

TABLE I. DATABASES & YEAR WISE PAPER DISTRIBUTION

Electronic Databases	**Total Papers Found**	**2015**	**2017**	**2018**	**2019**	**2020**	**Total**
Wiley Online Library	55	0	0	2	3	1	6
IEEE Explore	30	1	3	3	4	3	s14
Google Scholars Articles	532	0	1	0	4	3	8
ACM	22	0	1	2	0	0	3
Microsoft Academic	9	0	1	1	0	0	2
Total		**1**	**6**	**8**	**11**	**7**	**33**

TABLE II. QUALITY ASSESSMENT CRITERIA (QAC)

Sr. No	**Assessment Question (AQ)**	**Score**
AQ1	Is the study aim clear?	4
AQ2	Are the experimental datasets clearly stated?	4
AQ3	Are the used features clearly stated?	4
AQ4	Is the model clearly stated?	4
AQ5	Are empirical experiments clearly stated?	4
	Total Assesment Marks	**20**

TABLE III. QUALITY ASSESSMENT CRITERIA FOR SELECTED PAPERS AND INFORMATION EXTRACTION

Paper Title	**YOP**	**Methods used**	**Accuracy**	**QAC Score**
Logistic regression for polymorphic malware detection using ANOVA F-test [24]	2017	LR+ Polymorphism	97.7%	18
Android Malware Detection Based on Logistic Regression and XGBoost [25]	2019	LR+ Android API Call Sequence	96.18%	17
Identifying malicious botnet traffic using logistic regression [26]	2018	LR+ Botnet Traffic	97%	16
Droid-NNet: Deep Learning Neural Network for Android Malware Detection [27]	2019	Prediction for Malware using NN	0.988895% ±0.007	16
Machine Learning based Malware Detection in Cloud Environment using Clustering Approach [28]	2020	ML + Clustering Approach	94.3%	12
Online detection and control of malware infected assets [29]	2017	Malware activities in networks	Not Scalable	13
RHMD: Evasion-Resilient Hardware Malware Detectors [30]	2017	Hardware Malware Detectors	97.3%	14
Ensemble HMD: Accurate Hardware Malware Detectors with Specialized Ensemble Classifiers [31]	2020	Hardware Malware Detectors	95.8%	15
An Android Malware Detection Technique using Optimized Permissions and API with PCA [32]	2018	Android security at API level	97.25%	18
Permission based Android Malicious Application Detection using Machine Learning [33]	2019	Android Applications	99.34%	18
Malware Detection Modeling Systems [34]	2018	Malware Detection	Hypothetical Model	6

On Embedding Backdoor in Malware Detectors Using Machine Learning [35]	2019	Data Poisoning attack avoidance using backdoors	Reduce 30% Detection Rate in Malware	15
Quantifying the impact of adversarial evasion attacks on machine learning based android malware classifiers [36]	2017	SVM+LR	99.1%	14
A Knowledge-Domain Analyzer for Malware Classification [37]	2020	NB + SVM +LR	95.53 % using k-best	18
KerTSDroid: Detecting Android Malware at Scale through Kernel Task Structures [38]	2019	Android OS	NB+LR+SVM 96%	15
Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning [39]	2019	Cyber Supply Chain	LR+DT+SVM 66%	8
Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing [40]	2017	Security solutions for Virtual Infrastructure	98.84%	11
Machine Learning Based Intrusion Detection for Virtualized Infrastructures [41]	2018	ID's based Cloud Infrastructures	Acceptable level of Accuracy	4
Malware detection in android mobile platform using machine learning algorithms [422]	2017	Malware detection in Android Devices	DT+KNN+LR 20.56-91.69%	15
Securing Virtual Infrastructure in Cloud Computing using Big Data Analytics [43]	2018	Attacks on Virtual Infrastructure	Acceptable level of Accuracy	7
Android malware detection through generative adversarial networks [44]	2019	Malware detection using GAN	99%	18
Ransom ware detection using machine learning algorithms [45]	2019	Ransom ware detection using ML	90.27	14
Identification of Android malware using refined system call [46]	2019	NB+LR+SVM	99.9%	14
Voter Choice: A ransom ware detection honeypot with multiple voting framework [47]	2020	Ransom ware detection	98.6%	18
Deep neural network-based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities [48]	2020	Malware detection in IOT Network Traffic	97.01%	16
A machine learning framework for domain generating algorithm-based malware detection [49]	2019	Attacks during Communication of PCs	97.4%	16
Feature optimization and hybrid classification for malicious web page detection [50]	2020	Malicious Web Pages detection	95.76	17
Malware classification with recurrent networks [51]	2015	Malware detection using RNN	RNN+LR+ESN 98.3%	14
A Data Driven Characterization of Modern Android Spyware [52]	2020	Android malware attacks	97%	15
Routing Techniques in Data Center Network [53]	2015	Forwarding and routing techniques + DCN	Saves network traffic by 50 % and doubles throughput	15

TABLE 4. INFORMATION SYNTHESIS

Ref. No	Limitation	Future work
[24]	Worked on window XP and kali Linux only. There is a need to work and test on more platforms.	In future this method can likewise be tried on other working frameworks.
[25]	Their dataset is insufficient, only class and strategy name data are chosen as main feature.	XG-Boost and logistic regression should be applied to huge dataset and developed model should be improved to detect malware.
[26]	Integrated 8 different botnets traffic families to acquire good results. Better execution on Leave One Bot-Type traffic is needed.	They intend to apply more administered AI techniques, for example, SVM, RF and NN, to their dataset, and ultimately test the models across an enormous organization.
[27]	Doesn't Provide Limitations Information	Doesn't Provide Future Work Information
[28]	This methodology is limited in computational complexity and detection accuracy.	They intended to test the power of the heterogeneous systems and their convenience in the cloud climate by Assessing the model execution on a bigger dataset.
[29]	Study doesn't show any experimental proof.	They will try to focus on quantitative measures to recognize and control malware assets.

[30]	Malware detection is very difficult using simple detectors like logistic regression. They can easily avoid evasive attacks even after retraining because they know the base locators of detectors. They can be easily reversed engineered.	It is possible to avoid base locators, so they don't consider this case as a malware threat. Flexibility for this situation might be accomplished for future examination.
[31]	The integrated classifier gave restricted preferred position over a solitary general identifier.	In future they will utilize an alternate improvement method like Principal segment examination for dimensionality reduction and sifting techniques.
[32]	Highlight Filtering Methods and common used needs more improvement.	They will utilize another method like Principal part investigation for dimensionality reduction and will accomplish a good degree of exactness using powerful strategy of static API call.
[33]	Limited dataset by grouping only android application tests.	This work can be stretched out by considering different highlights of Android application like API Calls, Network exercises and so on. This dataset can be utilized for complex calculations to improve results.
[34]	Doesn’t provide restrictions	Doesn't Provide Information for Future Work
[35]	Doesn't Provide Information for restrictions	Threat will refine the detection technique intelligently for secondary passage malware to avoid security threat.
[36]	This methodology is restricted for bypassing static examination-based methodologies.	Classifiers that perform dynamic investigation will observe the first API calls whenever they are summoned. They will analyze static API call carefully using DREBIN.
[37]	Signature based malware detection methods is restricted with increasing number and type of malicious files due to unknown patterns.	They will design an integrated model for malware classification using different malware features like API calls and op-code sequences.
[38]	“Kertdroid” Data Provider have restricted assets for android records. They lead to lower accuracy rate while avoiding abnormal behavior with in-memory parallel data.	In future, they will change the information extraction method from paired to string for mathematical calculations.
[39]	Doesn't Provide Information for impediments	Profound learning-based methodologies will be considered in cyber chain systems (CCS). Also, will predict the future patterns in social site events.
[40]	Limitation of this methodology is the limited size of the CPT table for the Attack hub.	In future, they will increase the accuracy presented in zero-day attacks. They will analyze the computing instrument before developer address it.
[41]	This method is limited to only user application log and network log	The methodology is additionally irresistible against recently experienced malware attacks (Zero-day assaults).
[42]	Doesn't Provide Information for constraints	Future work will be the correlations of Random Forest and Support Vector Machine that can offer the best outcomes among the calculations analyzed.
[43]	Malware representation, grouping of viruses and demonstrating framework for malware attack.	They will design a framework for malware detection and increase the accuracy to detect zero-day attacks.
[44]	They didn't give the testing conditions to GAN algorithm due to lack of interlinked and fixed patterns. They did not provide inclusion metric of outcomes.	Their work can be stretched out in future while assessing different components in the payload of an application.
[45]	Doesn't Provide Information for constraints.	Doesn't Provide Information for Future Work
[46]	The proposed method could be compromised if an opponent is aware of the classification algorithm, parameters of classifier, and access to the dataset. In this case, he will infer the classification algorithm.	They will extend the developed approach using deep learning methods employing on multiple datasets. They intend to make the system robust by including attribute fusion approach.
[47]	The system has not been tested for the interception of executable traffic for drive-by download. The voting system failed with the dynamic detection method.	They will likewise prefer to incorporate other ransom ware types, for example, in PNG or PDF design later on work. They intend to extend voting system with multiple machine learning and deep learning methods for dynamic detection.
[48]	Doesn't Provide Information for restrictions	Doesn't Provide Information for Future Work
[49]	Domain name clustering algorithm being applied on known DGA domains only	In the future, they will extend this methodology using deep learning algorithms for domain name clustering and prediction of a real-world problems.
[50]	Number of malicious web samples is limited and the vector set consist of large number of features. Therefore, classifier led to poor performance due to excessive vector set.	In future, they will explore more new types of malicious webpage features, for example, the URL attribute features. In addition, they will focus on machine learning algorithms and will explore new methods for classification optimization.
[51]	It was difficult for the classifier to learn the language of malware because the classifier has to extract the useful information from the random temporal projection.	In future, recurrent modeling can be applied to the malware language modeling task to improve the results.

[52]	Static analysis of modern spyware is very weak against code obfuscation strategies while 20 % of samples crashed during dynamic analysis of spyware.	In future work, they will explore additional methods with respect to spyware to secure much experience such that designing new features, integrated dynamic analysis techniques etc.
[53]	Absolute bandwidth is limited between any server, lack of scalability, limited support for VMs migration etc.	In future, an independent topology will be designed for routing and forwarding. A complete and intelligent solution will be implemented for Load-balancing features.

Conclusion

With the huge proliferation of malware attacks day by day, it is very necessary to take preventive measures against them. Therefore after having an extensive Systematic Review, we conclude that traditional ML techniques along with deep learning techniques can be used to identify and detect malware in computer systems as well as in android devices. Each malware detection system has its own benefits and limitations. Studies reveals that Logistic Regression has many advantages over other Machine learning techniques in most of the scenarios. Logistic Regression Classifier is best to identify and detect malware in large and dispersed datasets. After doing comparative analysis on table 3 and 4, we conclude that Logistic Regression performs very well in securing Virtualized cloud-based environment as well.

LR classifier has outperformed among all due to its probabilistic nature. Logistic Regression classifier along with the combination of Naive Bayes, Random Forest, and Support Vector Machine is used to detect and avoid malware attacks and has achieved the highest accuracy around 99.9%. LR classifiers not only used to secure the systems from malicious activities but also impacts on the time and cost of the legitimate user. Therefore, in possible problems we have considered Logistic Regression as the main character.

References

[1] P. Szewczyk and M. Brand, “Malware detection and removal: An examination of personal anti-virus software,” 2008.

✓ [2] H. R. Zeidanloo, F. Tabatabaei, and P. V. Amoli, “All About Malwares (Malicious Codes),” in Proceeding of the 2010 International Conference on Security & Management, Las Vegas Nevada, USA, vol. 2, July,-12-15, 2010.

[3] J. Mayer, “Government hacking,” in Journal of Yale Law vol. 127 (3), 2019.

[4] H. Sayadi, N. Patel, S. M. P. D, A. Sasan, S. Rafatirad, and H. Homayoun, “Ensemble learning for effective run-time hardware-based malware detection,” in 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, USA, June 2018.

[5] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, “A review on feature selection in mobile malware detection,” Digital Investigation, vol. 13, pp. 22-37, June 2015,

[6] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, “A survey on malware detection using data mining techniques,” in ACM Computing Survey library, vol. 50 (3), pp. 1-40, October, 2017.

[7] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, “Large-scale malware classification using random projections and neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, October 2013.

[8] Y. D. Mane, “Detect and deactivate P2P Zeus bot,” in 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), July 2017.

[9] P. Haritha and R. Puviarasi, “Identification of malicious nodes & paths to reduce packet loss in mobile ADHOC network with NS2 simulator,” in International Journal of Innovative Technology and Exploring Engineering (IJITEE, vol. 8(9S2), July 2019.

[10] D. Nethra Pingala Suthishni and G. P. Ramesh Kumar, “Intrusion detection analysis by implementing fuzzy logic,” in Indian Journal of Computer Science and Engineering, vol. 2 (1), February 2011.

[11] S. Singh and M. Gosain, “Detecting forgery in images using alphabetic ordering of extracted blocks,” in International Journal of Technology and Computing (IJTC), vol. 1 (1), October 2015.

[12] R. M. Gomathi, P. Ajitha, A. Sivasangari, and T. Anandhi, “A comprehensive end-to-end identification and prevention system for cross site scripting attack for effective communication,” in Journal of Green Eng., 2020.

[13] D. Kiwia, A. Dehghantanha, K. K. R. Choo, and J. Slaughter, “A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence,” in Journal of Computational Sciences, vol. 27, pp. 394-409, July 2018.

[14] Y. Aafer, W. Du, and H. Yin, “DroidAPIMiner: Mining API-level features for robust malware detection in android,” in International Conference on Security and Privacy in Communication Systems, pp. 86-103, 2013.

[15] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, “DroidMat: Android malware detection through manifest and API calls tracing,” Tokyo, Japan 2012.

[16] N. McLaughlin et al., “Deep android malware detection,” in Proceeding of the Seventh ACM on Conference on Data and Application Security and Privacy, pp. 301–308, March 2017. 2017.

[17] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, “Droid-Sec: Deep learning in android malware detection,” in ACM SIGCOMM Computer Communication Review, vol. 44 (4), pp. 371–372, October 2014.

[18] T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, “Detection technique for hardware Trojans using machine learning in frequency domain,” in 4th Global Conference of Consumer Electronics GCCE, pp. 185–186, Osaka Japan, 2016.

[19] Y. Li, J. Gao, Q. Li, and W. Fan, “Ensemble learning,” in Proceeding of Data Classification: Algorithms and Applications, 2014.

[20] R. C. Merkle, “A digital signature based on a conventional encryption function,” in Proceeding of Advances in Cryptology, vol. 293, pp. 369-378, Berlin, Heidelberg.

[21] S. J. Rao, “Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis,” in Journal of American statistical association, vol. 98, no. 61, pp. 257-258, 31 Dec 2011.

[22] K. Özkan, Ş. Işik, and Y. Kartal, “Evaluation of convolutional neural network features for malware detection,” in 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-5, 2018.

[23] L. Cen, C. S. Gates, L. Si, and N. Li, “A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code,” in Journal of IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 4, pp. 400-412, 1 July-Aug. 2015.

[24] B. J. Kumar, H. Naveen, B. P. Kumar, S. S. Sharma, and J. Villegas, “Logistic regression for polymorphic malware detection using ANOVA F-test,” in International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1–5, Mar 2017.

[25] L. Suhuan and H. Xiaojun, “Android Malware Detection Based on Logistic Regression and XGBoost,” in 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp. 528–532, October 2019.

[26] R. Bapat et al., “Identifying malicious botnet traffic using logistic regression,” in Proceeding of Systems and Information Engineering Design Symposium (SIEDS), pp. 266–271, April 2018.

[27] M. Masum and H. Shahriar, “Droid-NNet: Deep Learning Neural Network for Android Malware Detection,” in 2019 IEEE International Conference on Big Data (Big Data), pp. 5789–5793, Dec. 201.

[28] R. Kumar, K. Sethi, N. Prajapati, R. R. Rout, and P. Bera, “Machine Learning based Malware Detection in Cloud Environment using

Clustering Approach," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7, Jul. 2020.

[29] H. Cam, "Online detection and control of malware infected assets," in 2017 IEEE Military Communications Conference (MILCOM), pp. 701–706, Oct. 2017.

[30] K. N. Khasawneh, N. Abu-Ghazaleh, D. Ponomarev, and L. Yu, "RHMD: Evasion-Resilient Hardware Malware Detectors," in Proceeding of 50th Annual IEEE/ACM on Microarchitecture (MICRO), pp. 315–327, Oct. 2017.

[31] K. N. Khasawneh, M. Ozsoy, C. Donovick, N. Abu-Ghazaleh, and D. Ponomarev, "Ensemble HMD: Accurate Hardware Malware Detectors with Specialized Ensemble Classifiers," in Journal of IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 3, pp. 620-633, 1 May-June 2020.

[32] S. R. Tiwari and R. U. Shukla, "An Android Malware Detection Technique Using Optimized Permission and API with PCA," in Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 2611–2616, Jun. 2018.

[33] A. Kapoor, H. Kushwaha, and E. Gandotra, "Permission based Android Malicious Application Detection using Machine Learning," in International Conference on Signal Processing and Communication (ICSC), pp. 103–108, March 2019.

[34] R. Kumar and G. S., "Malware Detection Modeling Systems," in International Conference on Recent Trends in Advance Computing (ICRTAC), pp. 187–192, 2018.

[35] S. Sasaki, S. Hidano, T. Uchibayashi, T. Suganuma, M. Hiji, and S. Kiyomoto, "On Embedding Backdoor in Malware Detectors Using Machine Learning," in 17th International Conference on Privacy, Security and Trust (PST), pp. 1–5, August 2019.

[36] Z. Abaid, M. A. Kaafar, and S. Jha, "Quantifying the impact of adversarial evasion attacks on machine learning based android malware classifiers," in Proceeding of IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–10, Oct. 2017.

[37] O. P. Samantray and S. Narayan Tripathy, "A Knowledge-Domain Analyzer for Malware Classification," in International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–7, March 2020.

[38] X. Wang and C. Li, "KerTSDroid: Detecting Android Malware at Scale through Kernel Task Structures," in IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), pp. 870–879, December 2019.

[39] A. Yeboah-Ofori and C. Boachie, "Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning," in International Conference on Cyber Security and Internet of Things (ICSIoT), pp. 66–73, May 2019.

[40] T. Y. Win, H. Tianfield, and Q. Mair, "Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing," in Journal of IEEE Transaction on Big Data, vol. 4, no. 1, pp. 11–25, Mar. 2018.

[41] N. Krishnan and A. Salim, "Machine Learning Based Intrusion Detection for Virtualized Infrastructures," in International CET Conference on Control, Communication, and Computing (IC4), pp. 366–371, 2018.

[42] M. A. Ali, D. Svetinovic, Z. Aung, and S. Lukman, "Malware detection in android mobile platform using machine learning algorithms," in International Conference on Infocom Technologies and Unmanned Systems (ICTUS), pp. 763–768, 2017.

[43] A. Kumar Gupta et al., "Securing Virtual Infrastructure in Cloud Computing using Big Data Analytics," in Journal of ETRI, vol. 41, pp. 569–573, October 2019.

[44] M. Amin, B. Shah, A. Sharif, T. Ali, K. Kim, and S. Anwar, "Android malware detection through generative adversarial networks," in Proceeding of Emerging Telecommunication Technology, pp. 3675, July 2019.

[45] S. Il Bae, G. Bin Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," in Journal of Concurrency and Computation Practice and Experience, vol.32, June2019.

[46] D. Kumar, G. Radhamani, P. Vinod, M. Shojafar, N. Kumar, and M. Conti, "Identification of Android malware using refined system calls," in Journal of Concurrency and Computation: Practice and Experience, vol. 31, no. 20, pp. 5311, October 2019.

[47] C. Keong Ng, S. Rajasegarar, L. Pan, F. Jiang, and L. Y. Zhang, "VoterChoice: A ransomware detection honeypot with multiple voting Framework," in Journal of Concurrency and Computation: Practice and Experience, vol. 32, no. 14, pp. 5726, July 2020.

[48] [49] D. K. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," in Journal of Emerging Telecommunication Technology, vol. 32, no. 7, 2020.

[49] Y. Li, K. Xiong, T. Chin, and C. Hu, "A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection," in IEEE Access, vol. 7, pp. 32765-32782, 2019.

[50] W. Deng, Y. Peng, F. Yang, and J. Song, "Feature optimization and hybrid classification for malicious web page detection," in Journal of Concurrency and Computation: Practice and Experience, July 2020.

[51] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1916-1920, 2015.

[52] F. Pierazzi, G. Mezzour, Q. Han, M. Colajanni, and V. S. Subrahmanian, "A Data-driven Characterization of Modern Android Spyware," in Journal of ACM Transaction on MIS, vol. 11, no. 1, pp. 1-38, 2020.

[53] M.. Majid, M. F. Hayat, F. Z. Khan, M. Ahmed, N. Jhanjhi et al., "Ontology-Based System for Educational Program Counseling. Intelligent" in Journal of Automation & Soft Computing, vol. 30, no. 1, pp. 373–386, 2021.

[54] S. Habib, F. Bokhari, S. Khan, "Routing Techniques in Data Center Network," in S. Khan, A. Zomaya (eds) Handbook on Data Centers, pp. 507 – 532, Springer, New York, NY, March 2015.