Gainfully Using Machine Learning Algorithm in Enhancing the Efficacy of Evaluating Malware Detection System[1]

Rishit Garkhel

DOI: 10.37648/ijrst.v10i03.005

Received: 05th July, 2020; Accepted: 04th August, 2020; Published: 10th September, 2020

ABSTRACT

The malware is an executable program that is very dangerous for pc or laptops. Some malware examples are adware, ransomware, bot, keyloggers, viruses, trojan horses etc. The aggressive hike of malware is very risky in confidential data. The limitation of existing classification algorithms is their detection performance and blocking the malware from affecting the systems. So, it is important to create a machine learning algorithm that helps detect and remove malware. With the help of this technique, we can remove the malware. Our research is accompanied by some classification algorithms such as Naive Bayes, Random Forest, MLP classifier, Bagging, AdaBoost, etc. We have evaluated the algorithms by accuracy, precision. Recall, frequency measure and others. For evaluation, we have used the WEKA machine learning and data mining tool. After implementing various algorithms, we concluded that the best accuracy of 99.2% is achieved by random forest.

I. INTRODUCTION

The tremendous advancement of organizations and assets has extended the number of Internet clients during devices going from frameworks to embedded structures. This Internet availability has offered various help to the end-clients, similar to straightforward and quick correspondence. Nowadays, end clients can see the value in web-based organizations through an Internet-related devices like mobiles, tabs, etc. Furthermore, this extending number of Internet customers instituted the poisonous software engineers to encourage noxious applications or projects commonly called malware. A tremendous measure of malware has been found in the new years, as depicted in Fig 1.

[1] *How to cite the article:* Garkhel R., Gainfully Using Machine Learning Algorithm in Enhancing the Efficacy of Evaluating Malware Detection System, IJRST, Jul-Sep 2020, Vol 10, Issue 3, 38-46, DOI: http://doi.org/10.37648/ijrst.v10i03.005

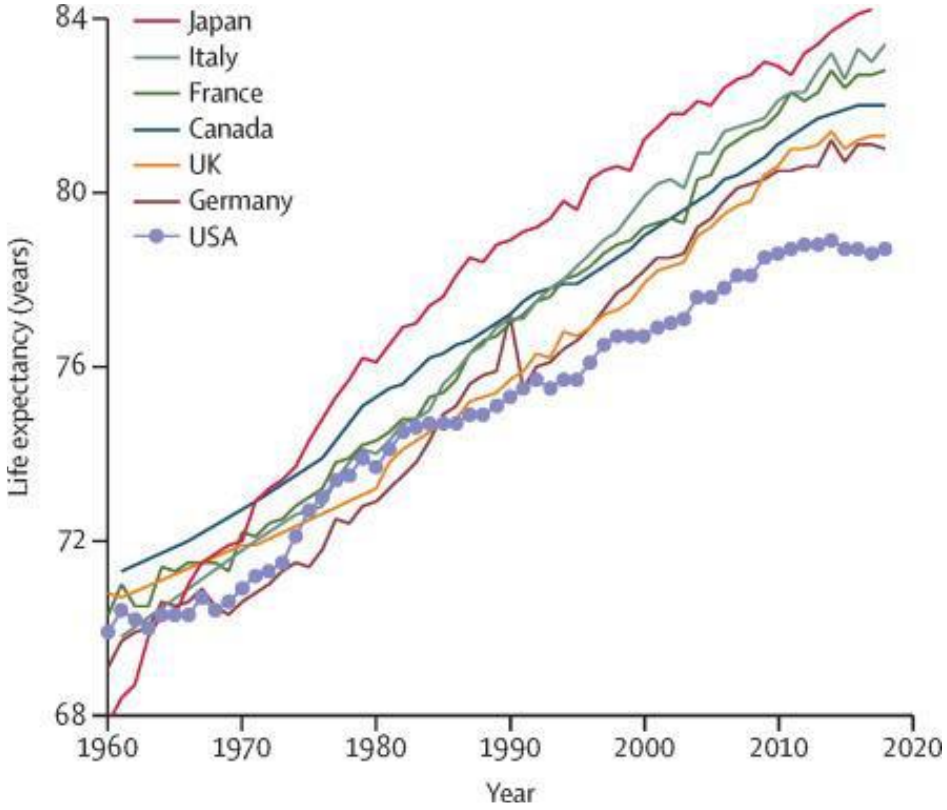

Fig 1: Rise in Malware detection

Various antivirus, interference area systems, and extra malware distinguishing proof structures have been created to keep away from hurt achieved by these malevolent undertakings. A couple of issues need quick thought because of the changing idea of malware and defect in existing programming. Various techniques from different controls have been anticipated for convincing malware revelation. Can sort the systems broadly into two orders, to be explicit, the static, signature-based processes and functional, direct based methodology. Static methods inspect malware reliant upon its plan, control stream, etc., without executing it. These strategies incorporate the establishment of an imprint informational index. The wide range of requirements is that existing techniques ignore to detect a common virus until it is trained. However, dynamic methods look at the malware tests through their execution. These strategies dismantle the lead of malware tests from their execution reports. Recently, malware software has been converting to be more complex and advanced using encryption systems. Static techniques disregard to recognize malware unequivocally. But fixed strategies have limited benefits compared to a dynamic system, because it is more active to hide the malware execution rights. Considering the advantages of emotional cycles, the point of meeting of recurring pattern investigate has moved to dynamic and automated methods for malware position.

Malware can be described as a destructive program that the customer inadvertently presents on their machine. Later on, these ventures can begin to agitate the appropriate action of the device or may continue undetected and do malignant exercises without being observed. When the aggressor manages the gadget, he would then move toward any information set aside on the machine. Some of the ambiguous strategies used to introduce malware on the PC structure during the website visit are repackaging the item, updating attacks, or painting for download. The attacker uses any of the techniques referred to before to make dangerous programming by embeddings a specific sort of malware into it before moving it to the web. Malware can be described as various programming that can release ruin on a PC structure or unlawfully use this information without the users' authorization. Some of the common malwares are botnet, trojans and ransomware. They are used to attack PC structures and perform crimes like a stunt, phishing, root access and system misuse.

Here, we evaluate the execution of representative AI strategies from different classes like decision tree-based, probability-based using a real malware dataset to the extent of an arrangement of execution estimations. Appraisal of ML methodology on various statures is critical because particular ML techniques have been expected to propel a substitute game plan of models. Like this, they can work in a relative environment.

In our research, we propose an intelligent combination of features extraction from implementation records of malware and present a great example. The main task of this research is:

• Extract effective malware examples from the virus by executing them in a cuckoo sandbox virtual environment.

• To create a real malware dataset, we collect the dynamic behaviour of features.

• Assessment of ML methods class-wise, for instance, decision tree-based, probability-based procedures, etc., to recognize the promising methodologies utilizing a real malware dataset.

• Experiential close to examining the ML strategies to perceive the greatest process for actual virus Identification to apply it as a contender methodology for mounting dynamic malware area structures.

II. SYSTEM

In this section, we discuss the overall system which is shown in figure 2.It joins dynamic malware revelation using ML technique involving data age stage, data extraction stage, grouping stage, and execution metric estimation stage. We used the Cuckoo sandbox environment for the execution of malware software and generates its execution report in JSON format. The data extraction stage removes features from JSON records that tend to the interesting behaviour of tests and denotes every model as obliging or malware. It delivers a veritable malware dataset used as a planning and test dataset by the grouping stage. The display metric computation stage registers malware area achieves terms of a combination of estimations.

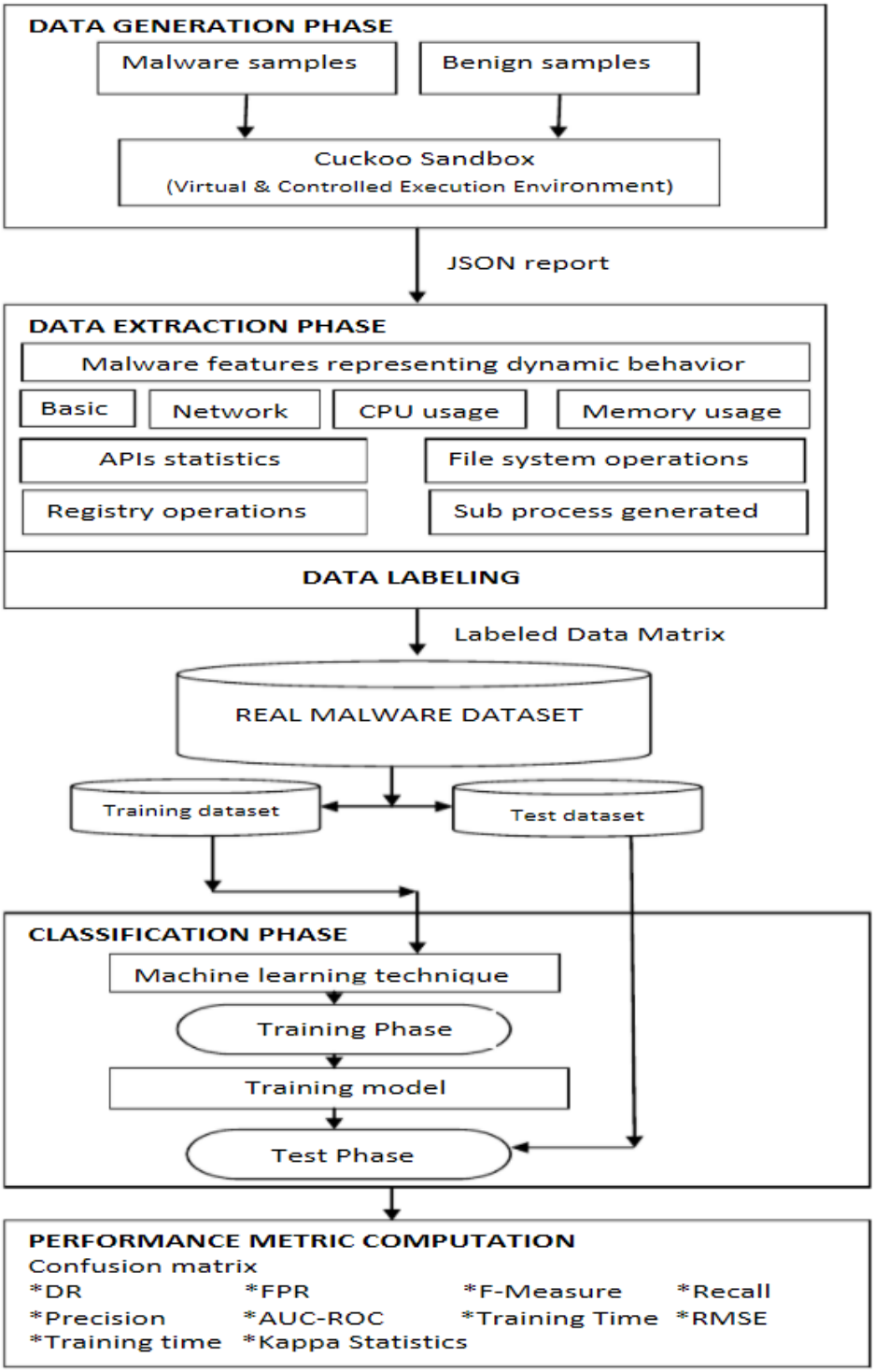

Figure 2: Evaluating of Malware Detection Proposed System

➢ **Data Generation Phase:** We have done a malware test using Cuckoo sandbox and Anubis at this stage. This environment grants malware execution and harmless doubles inside an isolated environment, analysing and recording their direct.

➢ **Data Extraction Phase:** The data accumulated through Cuckoo Sandbox, Anubi's will be accessible as JSON objects. The essential steps for the data extraction stage are as underneath:

1. Scrutinise spaces of JSON record

2. Concentrate features

3. malware test naming

➢ **Classification Phase:** A gigantic numeral of directed ML strategies have been planned to describe the dataset into many malware classes. For example, ANN is represented as the human brain for simulation. They can acquire any non-straight association among input and needed yield even inside seeing noisy planning data. Fascinated per client might explore the review of ML techniques referred to in the assessments. Lots of Machine learning algorithms have been implemented to classify malware classes in the dataset. For instance, ANN is created based on the human brain.

They can acquire any non-straight association among input and needed output even inside seeing noisy planning data. Fascinated per client might explore the review of ML techniques referred to in the assessments. For assessment purposes, we used WEKA and applied various Ml algorithm types that are inbuilt in that tool. We have taken 70% as training and 30% for testing in the dataset.

In our current research, we have applied the default setting of various machine learning available in weka. Nevertheless, the enhancement of points may improve the accuracy of the classification of machine learning methods. The trained model of the machine learning technique is well used to predict complicated malware groups. The output of this test includes the confusion matrix and other modulation. Security experts use the generated report for further execution measurement and understand strategy conclusion.

➢ **Performance Computation portion:** The presentation metric estimation stage ascertains the distinguished execution estimations from the disarray framework in the wake of the testing stage. The confusion matrix gives the possible improvement of Accuracy, f-measures, Recall in terms of FP, TN, and TP. It determines the weighted average of average execution algorithms like Accuracy, Recall, the false-positive rate. AUC-ROC types measurement is being used to decide measures.

III. ASSESSMENT MALWARE DATA

Here, we used VirusTotal for malware and similar detection. It is a website that analyses the URL entered by users, sorts it, and let them know that the requested URL contains a virus or trojan. VirusTotal clusters several antiviruses' details and online detection engines that check for infected URLs that say that the user's antivirus may not have detected or acknowledged unnatural links. Users can send up to 512MB of data to the site via email. Antivirus programming dealers can get copies of records hailed by different outputs yet passed by their engine to work on their product and, similarly, VirusTotal's ability. Customers can similarly examine suspected URLs and examined through the VirusTotal dataset. VirusTotal utilises Cuckoo sandbox for dynamic examination of malware. In the coming about the dataset, found a tremendous number of malware families. We requested malware tests into different families for evaluation inspiration driving ML systems, as indicated by their essential capacities.

Sr No	Class name	Number of samples
Training dataset		
1	Benign	3433
2	Trojan	4447
3	Virus	265
4	Worm	326
5	Packed	430
6	Backdoor	233
7	Hoax	41
8	DangerousObject	28
9	Adware	62
	Total	9265
Test dataset		
1	Benign	1451
2	Trojan	1889
3	Virus	115
4	Worm	140
5	Packed	206
6	Backdoor	109
7	Hoax	20
8	DangerousObject	13
9	Adware	28
	Total	3971

Table 1: Categories and Number of Samples in dataset

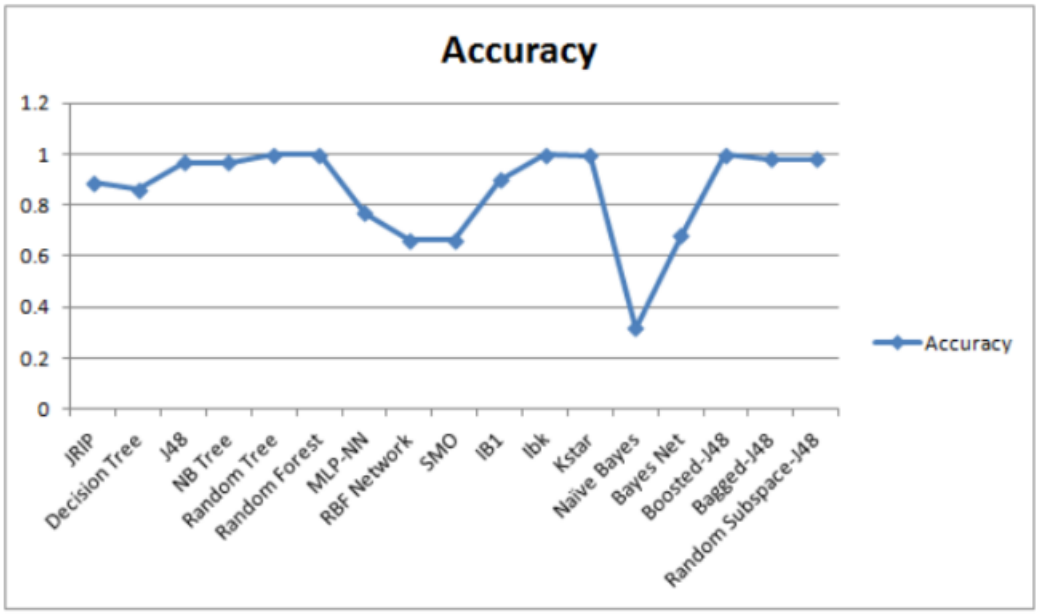

Figure 3: Performance comparison in terms of accuracy

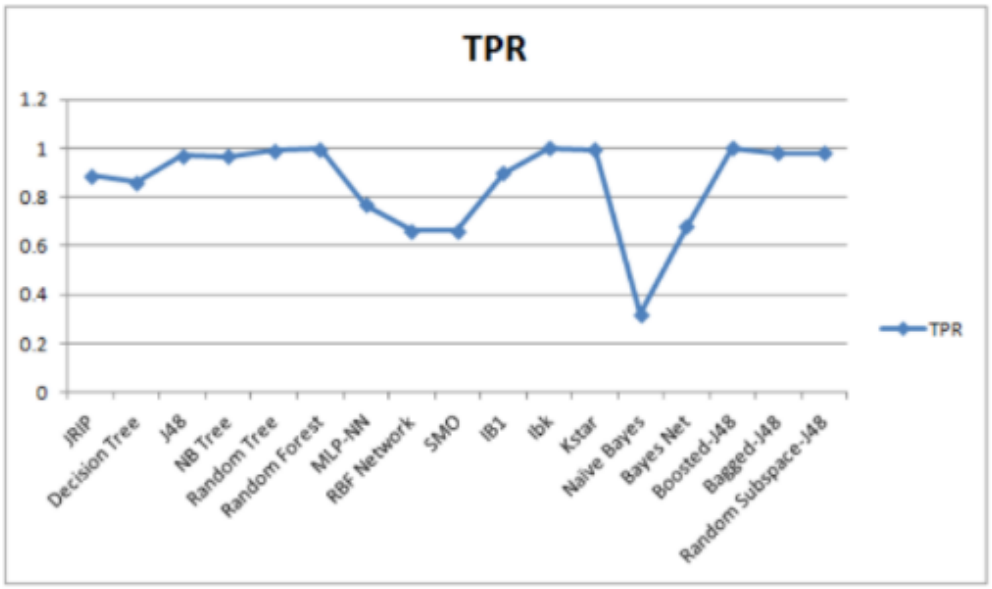

Figure 4: Performance comparison in terms of TPR

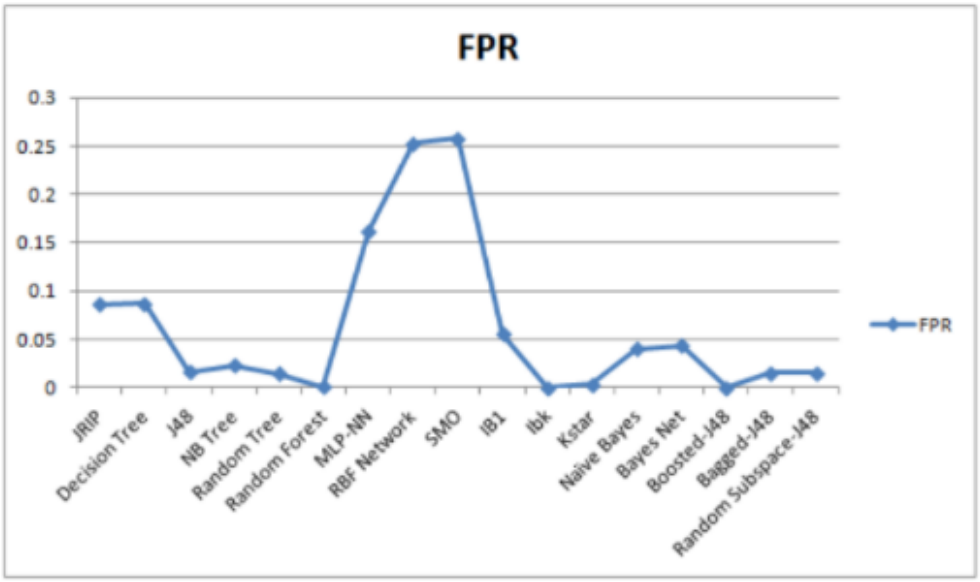

Figure 5: Performance comparison in terms of FPR

IV. CONCLUSION

In this work, an appraisal of managed ML strategies is done tentatively for perceiving malware utilising a veritable malware dataset to the extent of a variety of assessment estimations. The critical motivation driving using a grouping of evaluation estimations is that different procedures are made to progress unmistakable arrangement of models. A decent approach of elements has been separated from malware and generous executable models using a Cuckoo Sandbox and a Python-based automated system to shape a real malware dataset to assess ML procedures comprehensively. This work

recognises the best procedures for viable malware areas reliant upon an original mal-item dataset to the extent of distinguished execution measurements.

REFERENCES

[1]. SF Ahmad, SZ Ahmad, SR Xu, and B Li. Next generation malware analysis techniques and tools. In Electronics, Information Technology and Intellectualization: Proceedings of the International Conference EITI 2014, Shenzhen, China, 16-17 August 2014,page 17. CRC Press, 2015.

[2]. Ulrich Bayer, Andreas Moser, Christopher Kruegel, and Engin Kirda. Dynamic analysis of malicious code. Journal in Computer Virology, 2(1):67–77, 2006.

[3]. R. Bellman. Adaptive control processes: a guided tour Princeton university press. Princeton, New Jersey, USA, 1961.

[4]. Silvio Cesare and Yang Xiang. Software similarity and classification. Springer Science & Business Media, 2012.

[5]. Gianluca Dini, Fabio Martinelli, Andrea Saracino, and Daniele Sgandurra. Madam: a multi-level anomaly detector for android malware. In International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, pages240–253. Springer, 2012.

[6]. Manuel Egele, Theodoor Scholte, Engin Kirda, and Christopher Kruegel. A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys (CSUR), 44(2):6, 2012.

[7]. Christian Gorecki, Felix C Freiling, Marc K ¨uhrer, and Thorsten Holz. Trumanbox: Improving dynamic mal-ware analysis by emulating the internet. In Stabilization, Safety, and Security of Distributed Systems, pages 208–222. Springer, 2011.

[8]. Kent Griffin, Scott Schneider, Xin Hu, and TziCkerChiueh. Automatic generation of string signatures formal ware detection. In Recent advances in intrusion detection, pages 101–120. Springer, 2009.

[9]. Chun-Ying Huang, Yi-Ting Tsai, and Chung-HanHsu. Performance evaluation on permission-based detection for android malware. In Advances in Intelligent Systems and Applications-Volume 2, pages 111–120. Springer, 2013.

[10]. Youngjoon Ki, Eunjin Kim, and Huy Kang Kim. Anovel approach to detect malware based on api call sequence analysis. International Journal of Distributed Sensor Networks, 2015:4, 2015.

[11]. Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3):159–190, 2006.

[12]. G. Kumar and K. Kumar. Ai based supervised classifiers: an analysis for intrusion detection. In Proc. of International Conference on Advances in Computing and Artificial Intelligence, pages 170–174. ACM,2011.

[13]. G. Kumar and K. Kumar. An information theoretic approach for feature selection. Security and Communication Networks, 5(2):178–185, 2012.

[14]. G. Kumar, K. Kumar, and M. Sachdeva. The use of artificial intelligence based techniques for intrusion detection: a review. Artificial Intelligence Review,34(4):369–387, 2010.

[15]. Andreas Moser, Christopher Kruegel, and EnginKirda. Limits of static analysis for malware detection. In Computer security applications conference,2007. ACSAC 2007. Twenty-third annual, pages 421–430. IEEE, 2007.

[16]. S. Mukkamala and A.H. Sung. A comparative study of techniques for intrusion detection. In Proc. of 15thIEEE International Conference on Tools with Artificial Intelligence, 2003, pages 570–577. IEEE, 2003.

[17]. Fairuz Amalina Narudin, Ali Feizollah, Nor BadrulAnuar, and Abdullah Gani. Evaluation of machine learning classifiers for mobile malware detection. Soft Computing, 20(1):343–357, 2016.

[18]. Philip O'Kane, Sakir Sezer, and Keiran McLaughlin. Obfuscation: The hidden malware. Security & Privacy, IEEE, 9(5):41–47, 2011.

[19]. Konrad Rieck, Thorsten Holz, Carsten Willems, Patrick D¨ussel, and Pavel Laskov. Learning and classification of malware behavior. In Detection of Intrusions and Malware, and Vulnerability Assessment, pages 108–125. Springer, 2008.

[20]. Cuckoo Sandbox. Automated malware analysis, 2013.

[21]. Bhaskar Pratim Sarma, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. Android permissions: a perspective combining risks and benefits. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies, pages 13–22. ACM, 2012