A Survey on Malware Detection using Machine Learning in different Platforms.

Phd Scholar **Surbhi Prakash**, Professor **Dr. A.K Mohapatra**

Department of Information Technology Indira Gandhi Delhi Technical University Women

Abstract

Security threats have increasing drastically over the period. From virus, spyware, worm, trojan, ransomware to some many zero-day Malware is reported and exploited in different platforms. The platforms like Windows, Android, and Cloud (Iaas or Paas). The Phenomenon is like attacker always make target to humans via social engineering methodology or Phishing. When we talk about human the very first came in mind of attacker is the platform from which platform, they will be able to concentrate on the target. The basic approach used mostly in detection of malware in any platform is signature-based detection as that was quite beneficial but as malware are designed more in obfuscated manners, so it is quite difficult to detect those malicious activities using a signature-based approach. After signature-based approach, behavior-based approach is used for detection of malware. As some drawback appeared in both the approaches then researchers found the methodologies which can use Machine Learning Algorithms for example: KNN, Random Forest, Nearest neighbor etc.

Keywords: Malware Detection, Machine Learning, Hashing, Dataset.

1. INTRODUCTION

Despite all the improvements in cyber security still, malware is a persistent threat to web applications, mobile applications, or even cloud-based platforms. Malware analysis requests different techniques from several fields such as network and program analysis to understand their behavior and how they evolve. There is massive race between malware developers and analyst as malware developers tries to develop more and most complex malware so that it will be able to exploit for and for systems which will affect several organizations and even to common person in society, in the same manner analyst tries to make algorithm which will be able to detect those highly obfuscated malwares. For example, if any detector is constructed to detect the hashing pattern of any malware for example any malware signature consists of MD5 hash so it will be detected by more advanced techniques such as polymorphism or metamorphism.

According to AVT-Test Institute, 48 million malware samples were developed in the first quarter of 2017 [1]. As it was getting difficult for manual intervention in detecting the malware different techniques came into the picture which is Anti-virus software commonly uses a signature-based approach to detect the malware which provides less false-positive rate but in case of malware uses obfuscated code it will become quite difficult for signature-based approach to detect malware. On the other hand, behavior-based approach.

1.2 Method and Material

This section provide overview on malware types, basic malware detection techniques.

A. Malware Types

S.no	*Malware types*
1.	Virus
2.	Worms
3.	Trojan Horse
4.	Spyware
5.	Rootkit
6.	Ransomware
7.	Adware
8.	Botnet

Table 1: Malware types

B. Malware Detection Technique

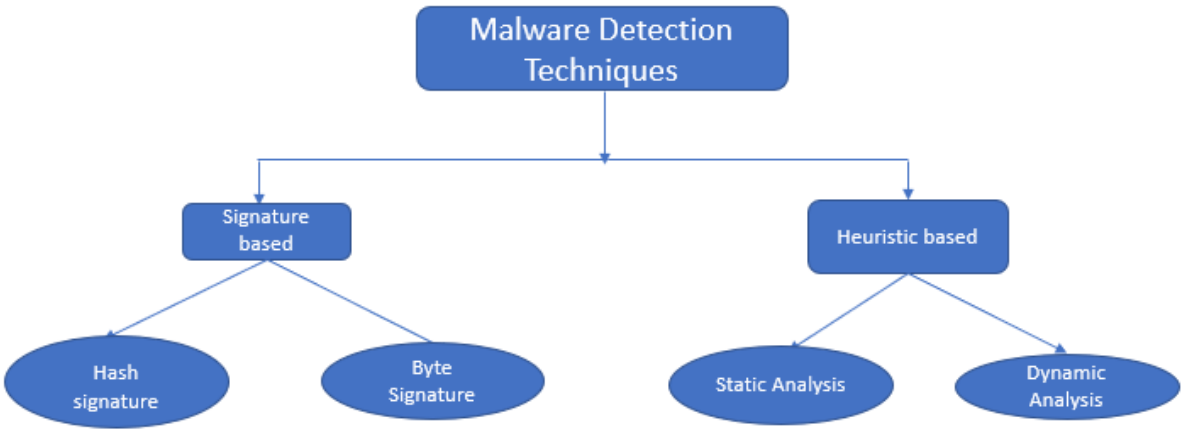

Fig 1: Malware Detection Technique

(i) Signature-based Approach.

The majority of Antivirus use a signature-based approach. This approach captures signature from the detected malware file and then use those signatures to find similar malware. A signature is basically a sequence of byte or hash values.

C. Heuristic-Based Detection

The heuristic-based approach also known as the behavior-based approach is used to detect malware not on the basis of signature but in use technique that possesses the ability to detect new malware as well but it has some drawbacks like it possesses a high false positive rate which detects malware.

Several techniques like Support Vector, Random Forest, and Naïve Bayes are used in behavior approaches.
Despite all the improvements in cyber security still, malware is a persistent threat to web applications, mobile applications, or even cloud-based platforms. Malware analysis requests different techniques from several fields such as network and program analysis to understand their behavior and how they evolve. There is a massive race between malware developers and analysts as malware developers try to develop more and more complex malware so that it will be able to exploit and for systems that will affect several organizations and even the common person in society, in the same manner, the analyst tries to make algorithm which will be able to detect those highly obfuscated malware. For example, if any detector is constructed to detect the hashing pattern of any malware for example any malware signature consists of MD5 hash it will be detected by more advanced techniques such as polymorphism or metamorphism

Static Analysis

Basic static analysis consists of examining the executable file without viewing the actual instructions. Basic static analysis can confirm whether a file is malicious, provide information about its functionality, and sometimes provide information that will allow you to produce simple network signatures. Basic static analysis is straightforward and can be quick, but it's largely ineffective against sophisticated malware, and it can miss important behaviors. A technique that only analyses Portable Executables without running it. To decompile executables some tools are used like IDA Pro, and OlleyDbg that display instructions, and provide information about malware.
All operating systems interact with API, In kernel32.dll there is Windows API "OpenFileW" that creates a new file or opens the existing file. As API calls reveal the behavior of the program and considered as major factor in the detection of malware Hashemi and Hamzeh presented a new approach that extracts unique opcodes from the executable file and converts them into digital images. Visual features are then extracted from the image using the Local Binary Pattern (LBP), which is one of the most famous texture extraction method in image processing. Finally, machine-learning methods are used to detect malware. The proposed detection technique obtained accuracy rate of 91.9% [23]. Shaid and Maarof also suggested

displaying malware in the form of images. Their technique captures API calls of malware and converts them into visual cues or images. These images are used to identify malware variants [24].

1) **Dynamic Analysis**

Compared to static analysis, dynamic analysis is more effective as there is no need to disassemble the infected file to analyze it. In addition, dynamic analysis is able to detect known and unknown malware. Furthermore, obfuscated and polymorphic malware cannot evade dynamic detection. However, dynamic analysis is time intensive and resource consuming [6].

2) **Hybrid Analysis**

Hybrid analysis is combination of static and dynamic analysis both. It fetch some functionality of Static analysis and some from dynamic analysis. Further, Ma et al.] introduced a method to reduce false positive in malware classification called Ensemble that combined static and dynamic classifier into one classifier. The method uses multi features include static import functions and dynamic call functions to improve the accuracy and reduce false positive. Furthermore, Santos et al. introduced OPEM, a tool to detect unknown malicious files by combining opcode frequency obtained during static analysis and system calls, operations and raised exceptions during dynamic analysis. OPEM showed accuracy of 95.9% from static analysis, 77.26% using dynamic and 96.6% using hybrid analysis with SVM.

3**) Dataset for static and Dynamic Analysis**

It is important to collect malware dataset for the researchers. One way to collect dataset is to capture responses from honeypot, malware dataset can be downloaded from anti-malware agents' websites such as Malware DB, Malwr, MalShare, VX Heaven, theZoo and VirusShare malware repository.Malware analysis in PE executable using Machine Learning

In this section or below sections emphasis is put on techniques for the modelling of malware detection in PE executable files.

Machine Learning Algorithm	Description
K-NN	-Nearest Neighbour (K-NN) classification algorithm classifies the input instance by considering the class label of k nearest training instances. The class of input instance is predicted as of the class of majority instances. Distance measures Euclidean, Manhattan, Hamming and Minkowski are used to find the class label of an input instance from nearest K nearest instances.
SVM	SVM algorithm creates a hyperplane to partitions the data instances of dataset input in different classes. For binary classification, a vector of points on two-dimensional input space can be visualized which separate the input data instance into two different classes benign class and malware class. Application of kernel function in SVM classifier training plays a vital role to classify the classes accurately. Linear, Radial and Poly kernel functions are commonly used in SVM classifiers.
LR	Logistic Regression is a parametric binary classification algorithm.LR learns the coefficients from the training data to build the logistic regression classifier. In general, LR estimates the empirical value of the parameter in a qualitative response model
DT	In decision tree classification, a decision tree is created by computing the info gain of each attribute in datasets. The attribute has maximum info gain becomes the root.

RF	Random Forest is an ensemble bagging machine learning algorithm. In DT only a single decision tree is created but in RF multiple decision trees are created based on independent subsets of the dataset with replacement. The outcome of random forest is computed through the votes given by every individual tree.

Table 2: Machine learning classification algorithms.

2) Malware Analysis in Android

Android is the most used operating system nowadays. Every specific website have their android application and can be downloaded from Google pay store as per availability of data easily it has some drawbacks as well like if some malicious user inject some malware in those applications and those will be downloaded by legitimate users and there whole mobile devices got compromised or infected. More than 3.25 million Android app that were infected till 2018.

3) Machine Learning approach for Android malware detection

Machine learning approaches is based on the learning method, which is typically divided into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning makes use of a labeled dataset of samples or instances to train the predictive model, which is often used to solve classification or regression problems.. The purpose of this kind of machine learning is to discover the internal structure or distribution characteristics of the datasets themselves, and it is often applied to problems such as data clustering and feature dimension reduction. Semi-supervised learning combines elements of supervised learning and unsupervised learning, using both labeled and unlabeled data.

There are physical implementation of Machine Learning in several projects:

1) Abstract the problem to be solved
2) Sample data acquisition and analysis
3) Data preprocessing
4) Feature selection
5) Model selection and training
6) Model evaluation
7) Use the new Dataset
8) Evaluation of the machine learning method to learn its performance

A) Sample Acquisition
B) Data Preprocessing
C) Feature Selection

3.1) Detection Evaluation

Evaluating the performance of Machine Learning Models is quite important in case of detecting Android malware it is quite more important.

A) DIVISION OF DATASET

The Original dataset is divided into training set and test set.
Training set is used to tune the model and tune the parameters.
Test set is used to evaluate the performance of classifier.

B) **CLASSIFIER PERFORMANCE**

In this section introduces performance metric which are used in android malware detection.

Predicted Results		
True Class	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 3: Confusion Matrix

The concepts of FP, FN, TP, and TN are defined as follows.

(1) True positive (TP): the application is a malicious application and was correctly predicted to be malicious;

(2) False positive (FP): the application is not a malicious application but was wrongly predicted to be malicious; (3) True negative (TN): the application is not a malicious application and was correctly predicted to be non-malicious; (4) False negative (FN): the application is a malicious application but was wrongly predicted to be non-malicious.

Some commonly used metrics are:

Accuracy (Acc): It depicts the correct predictions among the total number of samples.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Error rate (Err): It depicts ratio of false presentation among total number of samples.

$$Err = \frac{FP+FN}{TP + TN + FP + FN}$$

Precision (P) = Ratio of all sample correctly predicted to be positive among all the sample

$$P = \frac{TP}{TP + FP}$$

Recall (R) = represents the ratio of all positive samples correctly predicted among all positive samples

S.No	Resources	Results
1.	Combining file content and file relations for cloud-based malware detection.	Accuracy of system outperforms other popular antimalware software.
2.	Machine Learning based Malware Detection in Cloud Environment using Clustering Approach	It achieves better performance results when it is compared to the non-clustering approach in terms of accuracy, FPR, and AUC
3.	Cloud based malware detection for evolving data streams.	It achieves better detection accuracy than other stream data classification techniques
4.	SplitScreen: Enabling efficient distributed malware detection	The run time and memory usage of SplitScreen decreases as the number of signatures increases
5.	CAS: A framework of online detecting advance malware families for cloud based security	It indicates that CAS can detect high amount of malware samples efficiently at inline speed
6.	Mobile malware security challeges and cloud based detection	It is a promising Method for mobile security
7.	CloudEyes: Cloud -based malware detection with reversible sketch for resource -constrained internet of things (IoT) devices	It outperforms other systems with less time and communication consumption.
8.	Analyzing and optimizing cloud-based antivirus paradigm	It investigates the vulnerabilities and limitations of the cloud

Table 4: Cloud based detection techniques.

3)Proposed Framework and Conclusion

As we have done survey for Malware detection in various platforms like Windows, Android and cloud based. As per survey from different researchers approximately 560,000 new pieces of malware detected per day. Now it's going to be more than 1 billion. There are so many different approaches for detecting malwares in different platforms for example signature-based approach, behavior-based approach, and heuristic based approach. Later researchers have static analysis and dynamic analysis approaches. Later for better accuracy of the detection Machine Learning came in the picture then different algorithm is used to detect the malware in several platforms. Such as Artificial intelligence, deep learning, SVM, random forest, Naïve bayes etc. As per the survey individual approach was taken for detecting malware in several platform like only study done on windows single study done on Android and same single study done on cloud. In all the platform same techniques have been implemented and different results are obtained. As compared to all three platform most work till now done on Windows executables then on Android and lastly less on Cloud environment. There is another problem defined like algorithms in every platform is not yet confident to detect zero-day attack .

As a drawback of this survey which we observed Is like there no framework designed till now which will work on whole three platform to detect the vulnerabilities and malwares. So a proposed framework includes like collecting the samples of malware of different platforms like windows, Android and cloud as well. Then analysis started whether static or dynamic then feature extraction done after that training malware classifier then it went to Machine Learning classifier and lastly, we obtain the benign and Malicious file.

Next problem is for detection zero day attack so our proposed methodology will work on Adversarial malware analysis approach to detect the zero day malware with high accuracy.

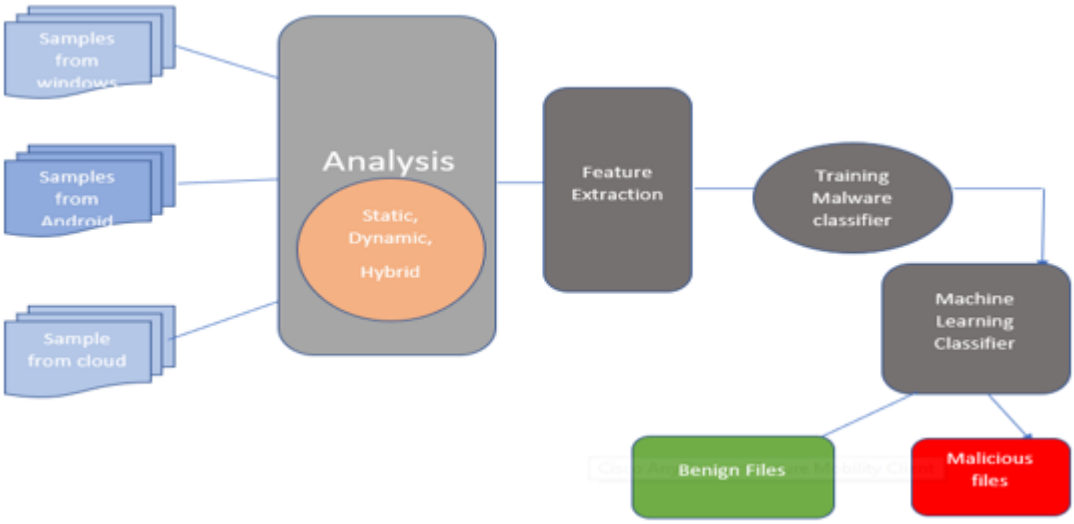

Fig 2: Proposed design

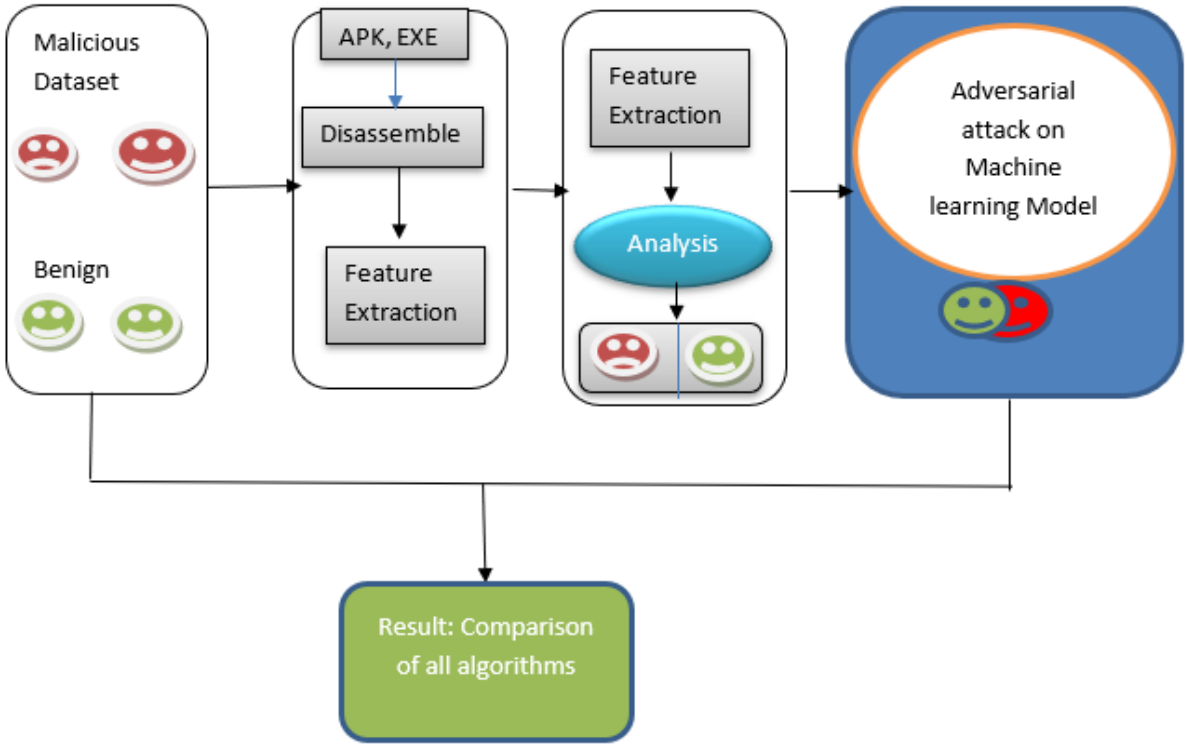

Fig 3: Proposed Framework using adversarial approach

References

[1] R.Mosli, R. Li, B. Yuan, and Y. Pan, "Automated malware detection using artifacts in forensic memory images," in 2016 IEEE Symposium on Technologies for Homeland Security, HST 2016, 2016, pp. 1–6.

[2] M.Karresand, "Separating Trojan horses, viruses, and worms - A proposed taxonomy of software weapons," in IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003, pp. 127–134.

[3] Smartphone Market Share. Accessed: Apr. 30, 2020. [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os

[4] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, ''Significant permission identification for Machine-Learning-Based Android malware detection,'' IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3216–3225, Jul. 2018

[5] M. Taleby, Q. Li, M. Rabbani, and A. Raza, ''A survey on smartphones security: Software vulnerabilities, malware, and attacks,'' Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 10, pp. 30–45, 2017

[6] A. A. A. Samra, H. N. Qunoo, F. Al-Rubaie, and H. El-Talli, ''A survey of static Android malware detection techniques,'' in Proc. IEEE 7th Palestinian Int. Conf. Electr. Comput. Eng. (PICECE), Mar. 2019, pp. 1–6

[7] . K. Gyamfi and E. Owusu, ''Survey of mobile malware analysis, detection techniques and tool,'' in Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON), Vancouver, BC, Canada, Nov. 2018, pp. 1101–1107.

[8] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, ''Android security: A survey of issues, malware penetration, and defenses,'' IEEE Commun. Surveys Tuts., vol. 17, no. 2, pp. 998–1022, 2nd Quart., 2015

[9] M. L. Polla, F. Martinelli, and D. Sgandurra, ''A survey on security for mobile devices,'' IEEE Commun. Surveys Tuts., vol. 15, no. 1, pp. 446–471, 1st Quart., 2012.

[10] Web of Science. Accessed: Apr. 30, 2020. [Online]. Available: https:// webofknowledge.com

[11] ccessed: Apr. 30, 2020. [Online]. Available: https://ieeexplore.ieee.

[12] SpringerLink. Accessed: Apr. 30, 2020. [Online]. Available: https://link. springer.com

[13] H. Zhang, S. Luo, Y. Zhang, and L. Pan, ''An efficient Android malware detection system based on method-level behavioral semantic analysis,'' IEEE Access, vol. 7, pp. 69246–69256, 2019

[14] S. Lou, S. Cheng, J. Huang, and F. Jiang, ''TFDroid: Android malware detection by topics and sensitive data flows using machine learning techniques,'' in Proc. IEEE 2nd Int. Conf. Inf. Comput. Technol. (ICICT), Kahului, HI, USA, Mar. 2019, pp. 30–36.

[15] M. Lindorfer, M. Neugschwandtner, and C. Platzer, ''MARVIN: Efficient and comprehensive mobile app classification through static and dynamic analysis,'' in Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf., Jul. 2015, pp. 422–433.

[16] Z. Ma, H. Ge, Y. Liu, M. Zhao, and J. Ma, ''A combination method for Android malware detection based on control flow graphs and machine learning algorithms,'' IEEE Access, vol. 7, pp. 21235–21245, 2019.

[17] T. Gao, W. Peng, D. Sisodia, T. K. Saha, F. Li, and M. Al Hasan, ''Android malware detection via graphlet sampling,'' IEEE Trans. Mobile Comput., vol. 18, no. 12, pp. 2754–2767, Dec. 2019

[18] A. N. Mucciardi and E. E. Gose, ‘‘A comparison of seven techniques for choosing subsets of pattern recognition properties,’’ IEEE Trans. Comput., vol. C-20, no. 9, pp. 1023–1031, Sep. 1971.

[19] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, MalDAE : Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics, Comput. Secur. 83 (2019)208–233, http://dx.doi.org/10.1016/j.cose.2019.02. 007.

[20] Y. Gao, Z. Lu, Y. Luo, Survey on malware anti-analysis, in: 5th International Conference on Intelligent Control and Information Processing, ICICIP 2014 - Proceedings, 2015, pp. 270–275, http://dx.doi.org/10.1109/ICICIP. 2014.7010353.

[21] J. Singh, J. Singh, Ransomware: an illustration of malicious cryptography (2) (2019), 1608–1611, http://dx.doi.org/10.35940/ijrte.B2327.078219.

[22] W. Zhang, H. Wang, H. He, P. Liu, DAMBA: Detecting android malware by ORGB analysis, IEEE Trans. Reliab. 69 (1) (2020) 55–69, http://dx.doi.org/10. 1109/TR.2019.2924677.

[23] J. Singh, J. Singh, J. Singh, Assessment of supervised machine learning algorithms using dynamic API calls for malware detection assessment of supervised machine learning algorithms using dynamic API calls for malware detection, Int. J. Comput. Appl. (2020) 1–8, http://dx.doi.org/10.1080/1206212X.2020. 1732641.

[24] M. Rabbani, Y.L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, P. Hu, A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing, J. Netw. Comput. Appl. 151 (2020) 102507, http://dx.doi. org/10.1016/j.jnca.2019.102507

[25] Wikipedia, Malware, 2020, https://en.wikipedia.org/wiki/Malware.

[26] Omer Aslan and Refik Samet, “A comprehensive review on malware detection approaches,” IEEE Access 8, 6249–6271, 2020.

[27] Deepti Gupta, Smriti Bhatt, Maanak Gupta, Olumide Kayode, and Ali Saman Tosun, “Access control model for google cloud iot. In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity),” IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). IEEE, 198–208, 2020.

[28] Wei Yan, “CAS: A framework of online detecting advance malware families for cloud-based security,” In 2012 1st IEEE International Conference on Communications in China (ICCC). IEEE, 220–225, 2012.

[29] Qublai K Ali Mirza, Irfan Awan, and Muhammad Younas, “A CloudBased Energy Efficient Hosting Model for Malware Detection Framework,” In 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 1–6, 2018.

[30] Deepti Gupta, Olumide Kayode, Smriti Bhatt, Maanak Gupta, and Ali Saman Tosun, “Learner’s Dilemma: IoT Devices Training Strategies in Collaborative Deep Learning,” In 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). IEEE, 1–6, 2020

[31] Jagsir Singh, Jaswinder Singh “ A survey on machine learning-based malware detection in executable files” Journal of Systems Architecture(Elsevier).