Challenges in Malware Detection and Effecting Areas: Survey

Gaurav Mehta, Prasenjit Das, and Vikas Tripathi

Abstract Malware detection is big area in domain of computer science and is never ending chase between malware scholars and security analyzer. Data mining is one of the favorite model for researchers to detect and classify malware in different areas like windows, android, and IoT. As the malware attacking technique and its ability to hide with obfuscation technique is changing rapidly, same is for detection method to detect malware with high accuracy and less time. On top of different method and techniques, the focus of detection process is shifting from binary and executable files to grayscale image and colored image analysis for detection. This paper focuses on different detection techniques, classification techniques, framework, dataset, and tools used by many researchers.

Keywords Malware detection · Image processing · Neural network · Data mining · IoT · Cloud · Android

1 Introduction

Threat of malicious code or malware is increasing at high rate due to growth of Internet and open-source platform like android. Currently, scope of malware is not only restricted to machines (desktop or laptops or mobile phones) but its existence can also be seen in IoT and cloud. The growth of IoT devices and cloud architecture had given big platform to malware detector to proceed for security breach and get personal information without the knowledge of host [1]. Millions of apps are available

G. Mehta (✉)
Department of Computer Science and Engineering, Chitkara University, Rajpura, Himachal Pradesh, India
e-mail: gaurav.mehta@chitkarauniversity.edu.in

P. Das
Department of Computer Application, Chitkara University, Rajpura, Himachal Pradesh, India
e-mail: prasenjit.das@chitkarauniversity.edu.in

V. Tripathi
Department of Computer Science and Engineering, Graphic Era, Dehradun, India

V. Goar et al. (eds.), *Advances in Information Communication Technology and Computing*, Lecture Notes in Networks and Systems 392,
https://doi.org/10.1007/978-981-19-0619-0_9

on Google Play Store and with millions of download count of thousands of apps which shows the popularity of android platform all over the world. In comparison to iOS platform, android platform also allows the users to download apps from unsecured or unverified links which increases the chance of attack on user device. Huge amount of android devices allows the attackers to target this platform, and 97% of attackers has the target field, i.e., android [3]. Each type of attacking malicious code had 50 different variants that make it more difficult to be identified by malware detection community [4]. Different approaches for static and dynamic analyses are used by researched to address security concerns. Malware detection is never ending process; it is a never ending chase between malware detector and malware creator [5]. Malicious code is not an emerging or new trend; it is from ages, since the start of computer machine. Large number of malware gets introduced in one or other fields which increase the demand to detect malware in all the areas that are attacked by malware. There exists large number of attacks which affects the host machine or data or security settings in one or the other way. Table 1 discusses some of the attacks and their types.

1.1 Malware Detection Approach

Anomaly-based malware detection: It is used to detect malicious activity both in network and computer. It is a process to detect malicious activity by comparing description of code. The classification of malicious code in anomaly-based detection is as per heuristic or rules based rather than signature or pattern. Major disadvantage of this technique is that little deviation from normal traffic or pattern gives alarm to security administrator to check and validate accordingly.

Signature-based malware detection: Database of known malwares is updated by malware detector, whenever new malicious code is identified. The signature of malicious code is added in this database to refer for malware detection. The new identifier is established for known threats to be identified in future. Signature-based technique has two major disadvantages: firstly, malware detection product/tool need to look into big database to identify attack; secondly, newly developed malware can't be detect by this approach [7].

Machine learning-based malware detection: Is a data analytics tool to effectively perform specific task. ML practice to detect malware/malicious code is considered by many researchers. The power of machine learning tools helps to differentiate malware from benign by using different classification and clustering algorithm.

2 Literature Review

In this section, we had mentioned comprehensive state-of-art for malicious code recognition and classification technique on the literature from year 2018 to 2020. Three stage frameworks have been proposed [8]: Stage 1—behavior of sample

files are extracted under scrutiny, and its interaction with OS is observed. In this stage, sample files are lope in sandbox environment—Virmon and Cuckoo. In stage 2—feature extraction is applied to analysis report, and label of sample is determined by virus total. In stage 3—dataset is divided into training set to get hold of testing and classification set to evaluate virus total—online multiple AV scan service. VirMon is used to extract Windows notification routines. Authors had used online machine learning framework—JUBATUS for malware classification based on behavior patterns. Lower feature space is achieved by using category-based modeling instead of API call-based modeling.

Mirza et al. [9] two main issues highlighted by authors are as follows: (1) Identifying malware accurately. (2) Enhance efficiency in term of energy for detection mechanism. CloudIntell uses ML technique to boost malware detection speed and support host methodology implementation based on cloud architecture. Weak classifiers performance is increased by decision trees, SVM, and then applying boosting on decision trees. Authors had developed automated feature extraction tool to extract features from 200,000 files. The tool also has the capability to remove obfuscated part of malicious file. Response and request queues configured using Amazons simple queue service (SQS) are monitored while forwarding the client request to detection engine.

Gu et al. [10] blockchain technologies are used to detect a mobile-based android malware for which framework CB-MMIDE ('Consortium blockchain for malware detection and evidence extraction') was proposed. Consortium chain by test members is compared with public chain by users in the consortium blockchain framework. Two features, i.e., permission information and signature are important features to be considered for malware detection; the consortium block chain framework is self-possessed of detecting consortium chain using test members public chain by users. In this work, feature modeling is performed to extract various features of malware families by statistical analysis method [11]. New malware gets introduced and that too in large number which makes the malware detection process to be more effective. Things get more critical when malware creators wrap the malware with techniques such as anti-emulation, packing, anti-virtualization, and obfuscation. Behavioral sequence chain is generated to collect malware followed by the process of clustering, preprocessing to create input sequence of MAS ('sequence alignment algorithm') which generate behavioral sequence chain of malware.

Chowdhury et al. [12] authors had used principal components analysis (PCA) to select features. The PCA has important feature of dimensionality reduction to enhance the computational speed. An ensembling of the API calls and n-gram features increases the effectiveness of malware detection. Integration of BAM and MLP neural network is proposed in which fast classification is achieved through BAM as it reduce dimensions of feature matrix [13]. Deep belief network (DBN) performs better as compared to support vector machines, decision trees, and k-nearest neighbor classifier algorithm. The machine language opcode describes the behavior of code/program. The opcode n-gram is used to describe the behavioral feature of malware as malware is represented as sequence of opcode. The model consists of PE parser, feature-extractor, and detection module for malware [14]. 'Convolution

neural networks' (CNNs) are used to detect malware based on image similarity. Binary code of malware is read as 8-bit unsigned integers to be organized in 2D array for visualization in grayscale image [0, 255] range. Images have large amount of dark spaces, and challenge is to find well-organized way to overcome weakness of NN that can be achieved by carefully analyzing binary file.

Wang et al. [15] authors had proposed network traffic analysis on multiple levels to identify features and combine it with machine learning algorithm. In this approach, HTTP and TCP network flow is monitored to determine the malicious activity. Data are collected under traffic collection module followed by feature extraction. The proposed framework includes foundation platform based on android virtual device, traffic generator to generate network traffic by installing and activating malware samples, traffic collector collect In/OUT bound network traffic and network proxy/firewall to analyze attack behavior.

Kim et al. [16] proposed model focuses on features like method opcode, string feature, API method, shared library function, permission and used component feature in detection process. Single feature vector is obtained by merging permissions, components, and environmental feature vector [17]. In proposed approach, multi-level fingerprint is extracted from application by n-gram analysis and feature hashing. These fingerprint features act as input to online classifier. The final decision on application to decide its benign or malware is based on confidence scores of classifier and device combination function. Feature of incremental learning of online classifiers helps to scale model for large number of applications to adapt it for new applications. Li et al. [18] proposed technique had used two features—permissions and API function calls which are used as input for the deep learning algorithm. The risky permissions and malicious API calls are combined to make feature set for weight-adjusted Droid-deep learning approach to distinguish benign from malware. Different features like APIs, permissions, IP address, and URL are packed in apk format to combine dangerous permission.

Kakisim et al. [19] features are extracted on the basis of descriptive and distinctive patterns of executables in isolated and virtual environment. Detection performance is increased by FS method at low dimension. Key observation of proposed work is that bi-gram increases as there is increase in samples.

AbRazak et al. [20] proposed bio-inspired algorithm approach to select permission features that are reliable and able to identify malicious code. Comparison of bio-inspired-algorithm PSO and to get finest features evolution, computation is done with information gain. ROC curve is used to visualize performance and gives reliable information of performance.

Ye et al. [21] proposed framework-heterogeneous deep learning is capable of detecting new malware. Framework is made of auto-encoder stacked up composed of multilayer restricted Boltzmann machines along with associative memory layer.

Detection process is divided in multiple steps as follows:

Step 1: Heterogeneous deep learning network is evaluated on labeled and unlabeled files with different parameters.

Step 2: Homogeneous deep network is compared with heterogeneous deep learning framework.
Step 3: Different shallow learning-based (ANN, SVM, NB, DT) classification methods are compared with proposed method.

Cai et al. [22] proposed dynamic app classification technique-DroidCat having dynamic feature set based on ICC intents and method calls. SOOT is used to transform apps (APK and SDK library) to Jimple code and run-time monitors to trace method call in Jimple code probes [23]. The most suitable option to detect android malware is SVM for binary files and KNN for manifest.xml files. Focused permissions for framework are users permissions, keywords from manifest.xml files, and strings from other files of applications. One combined feature vector is created by combining all the extracted features to achieve more accuracy. Karbab et al. [24] alDozer a supervised ubiquitous malware detection method that can be deployed both on server, IoT device and mobiles. Proposed works disassemble the class. DEX so as to produce VM assembly formalize to keep maximum raw information with minimum noise. Abdelsalam et al. [25] proposed malware detection method in cloud infrastructures using convolutional neural networks the 2D and 3D CNN approach. 2D CNN is employed by training on metadata of process in VM which is further enhanced by 3D CNN in which samples are collected during time intervals to reduce mislabeled samples during training [26]. MCSC proposed detection method as follows Step 1 opcode sequences are extracted from malware and then encode them with SimHash for equal length while preserving malware fingerprint. SimHash bits can be converted to images by taking each SimHash value as individual pixel. Step 2 CNN is adopted to train and identify the malware families. Converting malicious code to image and visualizing it to identify malware family is effective technique to detect malware. Malware classification using SimHash and CNN-MCSC approach is used to convert malware code to grayscale images using SimHash function to identify malware family by CNN.

Sharmeen et al. [27] proposed static, dynamic, and hybrid analysis methods based on features like suspicious permission list, API call list, and the system call list are identified. Features are extracted from apps (malware and benign) from different files like manifest, dex, byte code, and log files. To enhance the performance, accuracy, and detection rate, both static and dynamic features are used [28]. Three-level Hybrid model SAMDroid is proposed which combines the benefit of

(i) Static and Dynamic Analysis—improve analysis accuracy by combining benefits of both techniques.
(ii) Local and Remote Host—realistic inputs are taken from user during dynamic analysis.
(iii) Machine Learning Intelligence—remote host is used for detection operation to reduce memory overhead [29]. SIGPID has been proposed to extract significant permissions from side-to-side systematic pruning three-level approach by considering 22 permissions. (i) Permission ranking with negative rate (ii) Support-based permission ranking (iii) Permission mining with association rules is three major components for data pruning to reduce efforts required in analysis.

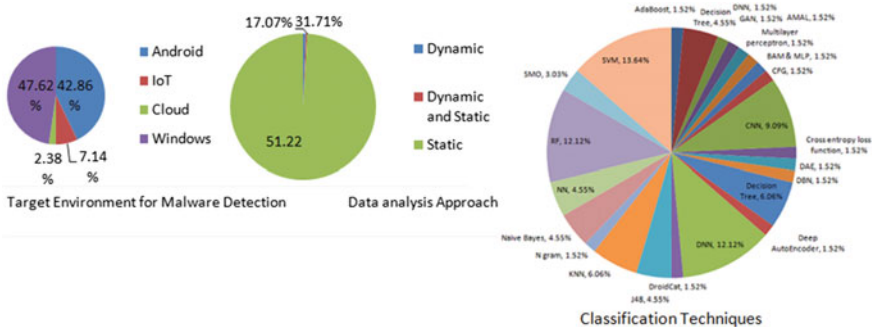

Fig. 1 Comparative malware target, analysis approach, and classification

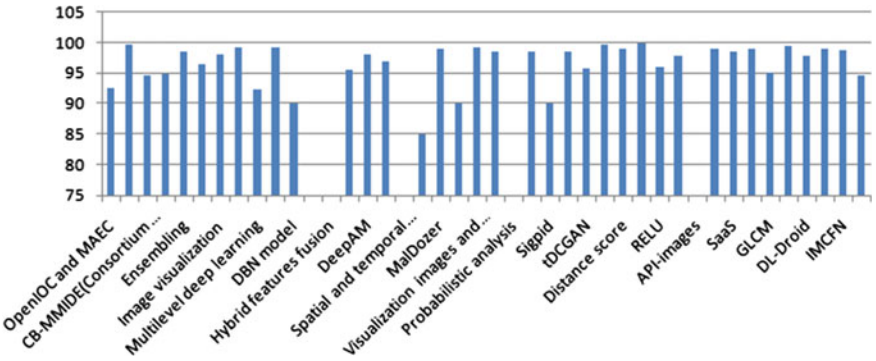

Fig. 2 Accuracy analysis of different methods

Venkatraman and Alazab [30] proposed work focus on feature and image-based visualization with similarity mining for identification and classification of malware. The technique is used to compare malware as per the behavior pattern and fast classification and detection of zero-day malware. Concept of visualization of the distance scores is used for malware detection. Classifiers (SVM and SMO) are used to compare results with four different kernels—normalized polynomial kernel, polynomial kernel, RBF, and PUK (Figs. 1, 2 and 3).

3 Conclusion

The paper presents literature review for different methods to detect malware in different fields like windows, android, IoT, and cloud. Papers were classified and investigated based on different approach like static or dynamic and on the basis of different classification technique as mentioned in Table 2. The detection of malware approach is reviewed on the basis of method used for malware detection, dataset

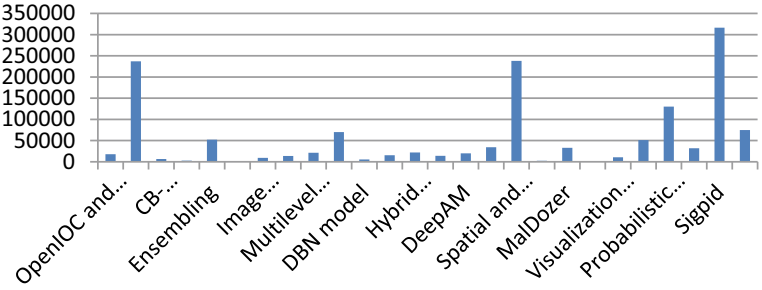

Fig. 3 Dataset analysis for different methods

used, total number of dataset values, accuracy, and case study of each method. The main idea of malware detection along with accuracy has been addressed for various methods used by researchers. Most of the article selected focus of malware detection accuracy by reducing detection time in different areas affected by malware like IoT, cloud, android, and windows. Figure 7 shows the different classification techniques used in different articles for malware detection, and Fig. 6 shows the accuracy for malware detection for various methods used in literature. OpenIOC and MAEC have 92.5%; CloudIntell has 99.69%; 'consortium blockchain for malware detection and evidence extraction' (CB-MMIDE) has 94.6%; API call sequence alignment and visualization have 94.89%; ensembling has 98.6%; opcode sequence has 96.5%; image visualization has 98%; traffic analysis has.99.3%; multilevel deep learning has 92.26%; feature hashing has 99.2%; DBN model has 90%; next-generation virus construction kit (NGVCK) has %; hybrid features fusion has 89.7%; bio-inspired algorithm has 95.6%; DeepAM has 98%; DroidCat has 97%; spatial and temporal perspectives have %, reverse engineered the android apps has 85%; MalDozer has 99%; IaaS cloud has 90%; ReLU has 96%; MALDAE has 97.89%; ScaleMalNet has 98.9%, SaaShas 98.5%; adaptive framework has 99.02%; GLCM has 95%; TrustSign has 99.5%; DL-Droid has 97.8%; IMCEC has 99%; IMCFN has 98.82%; KVMInspector has 94.7%. The review shows that malware detection is not only spotlight windows and android but also the upcoming field like IoT and cloud.

References

1. Saif, D., El-Gokhy, S.M., Sallam, E.: Deep belief networks-based framework for malware detection in android systems. Alex. Eng. J. **57**(4), 4049–4057 (2018)
2. IDC Research: Smartphone OS market share, 2015 q2. In: IDC Research Report (2015)
3. Kelly, G.: Report: 97% of mobile malware is on android this is the easy way you stay safe. In: Forbes Tech (2014)
4. Symantec: Latest intelligence for March 2016. In: Symantec Official Blog (2016)
5. Gibert, D., Mateu, C., Planes, J.: The rise of machine learning for detection and classification of malware: research developments, trends and challenges. J. Netw. Comp. Appl. 102526 (2020)

6. Taheri, R., Ghahramani, M., Javidan, R., Shojafar, M., Pooranian, Z., Conti, M.: Similarity-based android malware detection using Hamming distance of static binary features. Futur. Gener. Comput. Syst. **105**, 230–247 (2020)
7. Amin, M., Tanveer, T.A., Tehseen, M., Khan, M., Khan, F.A., Anwar, S.: Static malware detection and attribution in android byte-code through an end-to-end deep system. Futur. Gener. Comput. Syst. **102**, 112–126 (2020)
8. Pektaş, A., Acarman, T.: Classification of malware families based on runtime behaviors. J. Inform. Secur. Appl. **37**, 91–100 (2017)
9. Mirza, Q.K.A., Awan, I., Younas, M.: CloudIntell: an intelligent malware detection system. Future Gen. Comp. Syst. **86**, 1042–1053 (2018)
10. Gu, J., Sun, B., Du, X., Wang, J., Zhuang, Y., Wang, Z.: Consortium blockchain-based malware detection in mobile devices. IEEE Access **6**, 12118–12128 (2018)
11. Kim, H., Kim, J., Kim, Y., Kim, I., Kim, K.J., Kim, H.: Improvement of malware detection and classification using API call sequence alignment and visualization. Clust. Comput. **22**(1), 921–929 (2019)
12. Chowdhury, M., Rahman, A., Islam, R.: Malware analysis and detection using data mining and machine learning classification. In: International Conference on Applications and Techniques in Cyber Security and Intelligence, pp. 266–274. EdizionidellaNormale, Cham (2017)
13. Yuxin, D., Siyi, Z.: Malware detection based on deep learning algorithm. Neural Comput. Appl. **31**(2), 461–472 (2019)
14. Kumar, R., Xiaosong, Z., Khan, R.U., Ahad, I. and Kumar, J.: Malicious code detection based on image processing using deep learning. In: Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, pp. 81–85 (2018)
15. Wang, S., Chen, Z., Yan, Q., Yang, B., Peng, L., Jia, Z.: A mobile malware detection method using behavior features in network traffic. J. Netw. Comput. Appl. **133**, 15–25 (2019)
16. Kim, T., Kang, B., Rho, M., Sezer, S., Im, E.G.: A multimodal deep learning method for android malware detection using various features. IEEE Trans. Inf. Forensics Secur. **14**(3), 773–788 (2018)
17. Zhang, L., Thing, V.L., Cheng, Y.: A scalable and extensible framework for android malware detection and family attribution. Comput. Secur. **80**, 120–133 (2019)
18. Li, W., Wang, Z., Cai, J., Cheng, S.: An android malware detection approach using weight-adjusted deep learning. In: 2018 International Conference on Computing, Networking and Communications (ICNC), pp. 437–441. IEEE (2018)
19. Kakisim, A.G., Nar, M., Carkaci, N., Sogukpinar, I.: Analysis and evaluation of dynamic feature-based malware detection methods. In: International Conference on Security for Information Technology and Communications, pp. 247–258. Springer, Cham (2018)
20. AbRazak, M.F., Anuar, N.B., Othman, F., Firdaus, A., Afifi, F., Salleh, R.: Bio-inspired for features optimization and malware detection. Arab. J. Sci. Eng. **43**(12), 6963–6979 (2018)
21. Ye, Y., Chen, L., Hou, S., Hardy, W., Li, X.: DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowl. Inf. Syst. **54**(2), 265–285 (2018)
22. Cai, H., Meng, N., Ryder, B., Yao, D.: Droidcat: Effective android malware detection and categorization via app-level profiling. IEEE Trans. Inf. Forensics Secur. **14**(6), 1455–1470 (2018)
23. Rehman, Z.U., Khan, S.N., Muhammad, K., Lee, J.W., Lv, Z., Baik, S.W., Shah, P.A., Awan, K., Mehmood, I.: Machine learning-assisted signature and heuristic-based detection of malwares in android devices. Comput. Electr. Eng. **69**, 828–841 (2018)
24. Karbab, E.B., Debbabi, M., Derhab, A., Mouheb, D.: MalDozer: automatic framework for android malware detection using deep learning. Digit. Investig. **24**, S48–S59 (2018)
25. Abdelsalam, M., Krishnan, R., Huang, Y., Sandhu, R.: Malware detection in cloud infrastructures using convolutional neural networks. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 162–169. IEEE (2018)
26. Ni, S., Qian, Q., Zhang, R.: Malware identification using visualization images and deep learning. Comput. Secur. **77**, 871–885 (2018)

27. Sharmeen, S., Huda, S., Abawajy, J.H., Ismail, W.N., Hassan, M.M.: Malware threats and detection for industrial mobile-IoT networks. IEEE Access **6**, 15941–15957 (2018)
28. Arshad, S., Shah, M.A., Wahid, A., Mehmood, A., Song, H., Yu, H.: SAMADroid: a novel 3-level hybrid malware detection model for android operating system. IEEE Access **6**, 4321–4339 (2018)
29. Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., Ye, H.: Significant permission identification for machine-learning-based android malware detection. IEEE Trans. Industr. Inf. **14**(7), 3216–3225 (2018)
30. Venkatraman, S., Alazab, M.: Use of data visualisation for zero-day malware detection. Secur. Commun. Netw. (2018)