A REVIEW ON MULTIMODAL DEEP LEARNING METHOD FOR ANDROID MALWARE DETECTION

[1]AFSANA, [2]Dr. SAYED ABDULHAYAN

[1]M.Tech. Student, [2]Associate Professor
Department of TCE
DSCE, Bengaluru

***Abstract*: With the global use of smart phones, the number of malware has been increasing rapidly. Among all the smart devices, Android devices are the most targeted devices by malware due to their high popularity. This paper set forth a framework for Android malware detection and prevention. The various kinds of features are used by the framework to reflect the various properties of the Android applications from different aspects , and the features are refined using similar based or existence based of feature extraction method for feature representation of malware detection and prevention. For this a multimodal deep learning method is introduced and to be used as a malware detection and prevention model for Androids. With this detection model, it will be possible to gain the maximum benefits of multiple features. To estimate performance, various experiments are carried out and the accuracy of the model is the deep neural network models. Further the framework is evaluated in various types of aspects such as efficiency of model in updates usefulness of the diverse features and the representation of the feature method.**

***Keywords*: Machine learning Android malware, intrusion detection, malware detection, Neural network.**

I. INTRODUCTION

With the high popularity of mobile devices such as tablets or smart phones, attacks are increasing rapidly on mobile devices. Mobile malware is the most dangerous threats in technologies which cause various financial damages as well as security incidents. According to G DATA report [9] in 2017, security experts were discovered almost 750,000 Android malware during first 3months of 2017. It expected that a greater number of mobile malware will keep on developed and spread for committing various cybercrimes of mobile devices. Android is a most targeted mobile operating system that is used by mobile malware because of their high popularity of Android devices. In addition to number of Android devices, there is also another reason that increases malware authors for developing Android malware.

The main reason behind this is that the operating system of Android allows users to install the applications from third-party and attackers can mislead users of Android to download suspicious applications or malicious from attackers' servers.

To stop the attacks caused by Android malware, various researches have been proposed. The malware detection research approaches can be classified into two different categories; dynamic analysis based detection [1-8] and static analysis based detection [10-14]. The static analysis methods can be use for syntactic features that can be able to extracted without executing any application, where the dynamic analysis methods will be use for semantic features that can be able to monitored when any kind of an application can be executed in a controlled type of environment. Static analysis has the advantage that it is unnecessary to set the computational overheads and the execution environments, for static analysis are relatively less. Dynamic analysis has the advantage that it can be possible to handle applications which are malicious and use some of the obfuscate techniques such as code packing or encryption.

There are various previous works that related to Android malware detections, but more previous studies uses only limited type features to detect the malware. Each type of the feature can be able to represent only few properties of applications. On the other way, here a framework proposed to detect malware by using various feature of information to reflect many characteristics of applications in different aspects. The proposed framework will extracts at first and processes the multiple feature types, and remedy them using the feature of vector generation methods. The feature vector generation method is consists of similarity-based method and existence-based method and these methods are very much effective to differentiate between benign and malware applications though the malware has so many similar properties of benign applications. In addition to this the framework uses classification model that defines the degree of classification in respect to their importance. Among so many useful algorithms, the deep learning algorithm is concluded that it is suitable classification algorithm for the framework that uses various features types.

III. PROPOSED SYATEM

Here the multimodal deep neural network model is proposed to verify the features having different properties. The multimodal deep learning method is utilized to make neural network to refine the properties with different features. For example, as we can consider a multimodal deep learning method was used to recognize the human speech using mouth shape information and voice information [15]. Here the different types of features are input and then can be proceed in different neural networks

that to separately, and then each initial neural network is connected to the final neural network to produce classification results. According to survey, the research is first application of multimodal deep learning for Android malware detection.

Many experiments are conducted using the framework with large data from the known small dataset of the Malgenome project [16] and Virus Share [17]. Measured and compared performance of the model can be done with the help of the deep neural network model. In addition to this the evaluation of the framework in many aspects is include efficiency in various model updates, the diverse features are very much useful and effects the feature representation method. As compared to results with other deep learning methods, this framework is having the good performance in malware detection.

III. SOFTWARE USED

Detection and prevention of the malware attacks in android systems can be done by using the Machine learning in Python. Machine Learning using Python it is a type of the artificial intelligence (AI) that can provides for computers with the ability to learn without being explicitly programmed for the algorithm. Machine learning is focuses on the development of the Computer Programs that can be change when those are exposed to the new data.

ANACONDA NAVIGATOR

The anaconda navigator is rapidly developing Open Data Science platform which uses Python which can be considered as one of the fastest growing Open Data Science language.

Anaconda is also a free and open-source[18] distribution of a and R programming language and Python for the purpose of scientific computing (machine learning applications, data science, predictive analytics, large-scale data processing, etc.), this aims to the simplification of deployment and package management. Package management system conda use to manage Package versions [19]. Over 13 million users use the Anaconda distribution and it includes almost more than 1400 most used data-science packages that are suitable for Linux, Windows, and MacOS[20].

Jupyter is Scientific Python Development Environment which is the one of the most used application of the Anaconda navigator. It can be described and mentioned as a powerful most interactive development environment for Python language with the advanced features such as , interactive testing, editing , introspection and debugging features and a numerical computing environment, and it also includes popular Python libraries such as SciPy (signal and image processing) , NumPy (linear algebra), and matplotlib (interactive 2D/3Dplotting).

IV RELATED WORK

All previous approaches that were done using deep learning algorithm can be explained like , turn. Razvan Pascanu [21] detect the Windows based malware by using the recurrent neural network. They used the API events for the features in terms of detection. Deep Sign [22] used the dynamic API calls and also their parameters as features for the Windows based malware detection method. Here the deep network is used by the Deep Sign to classify benign and malware files. A deep neural network method is proposed by Joshua Saxe [23] for malware detection. In that method , strings, entropy , PE import functions , and metadata of Windows binaries were used as of features. Various kind of features all together used in this method , therefore this method seems to be difficult to detect the malware like Trojan , because it includes many features that of normal programs.

Droid detector [24], used machine learning method, to detect the Android malware. This method uses the extraction of various features and then those extracted features will be used in the deep network. Wei yu [25] set forth a method of Android malware detection system that used to model the neural network with the permissions and the system calls are traced from various applications. In that system, static features are used as only permissions , even though there is a lot of information that can be tested and used for the useful in detection. Niall McLaughlin [26] proposed a method for Android malware detection system that includes a convolution neural network (CNN).

In that system , without any refinement the raw opcode sequences of applications are used as the features. Hossein Fereidooni [27] proposed a system called as ANASTASIA, to detect the Android malware using various features such as permissions, intents, API calls and system commands. That system uses deep neural network and also many classifiers including that . Even though various kinds of features can be able to extracted from the Android applications, and also the most previous methods use small number of features in detection. In addition to this the previous methods are do not consider as the situations when these are adding to the new feature types.

CONCLUSION

This paper includes a literature review about the smart eye blink solution for MND patient, which by researched will be overcome with proposed method with greater accuracy and quick response compare to older techniques.

REFERENCES

[1] D. Luke, V. Notani, and A. Lakhotia, "Droidlegacy: Automated familial classification of android malware," In Proc. of the ACM SIGPLAN on Program Protection and Reverse Engineering Workshop, pp. 3, 2017.

[2] A. Zarni, and W. Zaw, "Permission-based android malware detection," International Journal of Scientific and Technology Research, vol. 2, no. 3, pp. 228-234, 2017.

[3] Ch.-Y. Huang, Y.-T. Tsai, and C-H. Hsu, "Performance evaluation on permission-based detection for android malware," Advances in Intelligent Systems and Applications, vol. 2, pp. 111-120, 2016.

[4] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket," In Proc. the Network and Distributed System Security Symposium (NDSS), vol. 14, pp. 23-26, 2016.

[5] D-J. Wu, C-H. Mao, T-E. Wei, H-M. Lee, K-P. Wu, "Droidmat: Android malware detection through manifest and api calls tracing," In Proc. of the Asia Joint Conference on Information Security (Asia JCIS), pp. 62-69, 2017.

[6] S. S. Hao, B. Liu, S. Nath, W. G. Halfond, and R. Govindan, "PUMA: Programmable UI-automation for Large-scale Dynamic Analysis of Mobile Apps," In Proc. of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 204-217, 2017.

[7] M. E. Chin, A. P. Felt, K. Greenwood, D. Wagner, "Analyzing inter-application communication in Android," In Proc. of the international conference on Mobile systems, applications, and services, pp.239-252, 2017.

[8] P. PF Chan, L. CK Hui, SM. Yiu, "Droidchecker: analyzing android applications for capability leak," In Proc. of the ACM conference on Security and Privacy in Wireless and Mobile Networks, pp. 125-136, 2015.

[9] A G DATA Report, "8,400 new android malware samples every day".

[10] W. Enck., P. Gilbert, S. Han, V. Tendulkar, B-G. Chun, L. P. Cox, J. Jung, P. Macdaniel, A. N. Sheth, "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," ACM Transaction on Computer Systems, vol. 32, no. 5, 2017.

[11] L. K. Yan, H. Yin, "DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis," In Proc. of the USENIX Security Symposium, pp. 569-584, 2016.

[12] T. Bläsing, L. Batyuk, A-D. Schmidt, S. A. Camtepe, S. Albayrak, "An android application sandbox system for suspicious software detection," In Proc. of the Malicious and unwanted software (MALWARE), pp. 55-62, 2017.

[13] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, Y. Weiss, "Andromaly: a behavioral malware detection framework for Android devices," Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161-190, 2016.

[14] A- D.Schmidt, F. Peters, F. Lamour, C. Scheel, S. A. Camtepe, S. Albayrak, "Monitoring smartphones for anomaly detection," Mobile Network sand Applications, vol. 14, no. 1, pp. 92-106, 2017.

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

[16] O. E. David, N. S. Netanyahu, "Deepsign: Deep learning for automatic malware signature generation and classification," In Proc. of the International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2015.

[17] J. Saxe, K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," In Proc. of the 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 11-20, 2015.

[18] Z. Yuan, Y. Lu, Y. Xue, "Droiddetector: android malware characterization and detection using deep learning," Tsinghua Science and Technology, vol. 21, no. 1, pp. 114-123, 2016.

[19] W. Yu, L. Ge, G. Xu, X. Fu, "Towards Neural Network Based Malware Detection on Android Mobile Devices," Cybersecurity Systems for Human Cognition Augmentation, pp. 99-117, 2018.

[20] N. Mchaughlin, J. Martinez del Rincon, B-J. Kang, S. Yerima, Y. Safaei, E. Trickel, Z. Zhao, A. Doupe, G. Joon Ahn, "Deep Android Malware Detection," In Proc of the ACM on Conference on Data and Application Security and Privacy (CODASPY), pp. 301-308, 2017.

[21] H. Fereidooni, M. Conti, D. Yao, A. Sperduti, "ANASTASIA: ANdroid mALware detection using STAtic analysis of Applications," In Proc. of the IFIP International Conference on New Technologies, Mobility and Security, pp. 1-5, 2016.

[22] APKtool. (September , 2017) [Online]. Available: https://ibotpeaches.github.io/Apktool.

[23] IDA pro. (September , 2017) [Online]. Available: https://www.hex-rays.com/products/ida

[24] Released Code (September, 2017) [Online]. Available: https://github.com/cloudio17/A-Multimodal-Deep-Learning-Method-for-Android-Malware-Detection.

[25] A. Yousra, W. Du, H. Yin, "DroidAPIMiner: Mining API-level features for robust malware detection in android," In Proc. of the International Conference on Security and Privacy in Communication Systems, pp. 86-103, 2015.

[26] Y. Bengio, "Learning deep architectures for AI," Foundations and Trends in Machine Learning, vol. 2, no. 1, 2017.

[27] T. Abou-Assaleh, N. Cercone, V. Keselj, R. Sweidan, "N-gram-based detection of new malicious code," In Proc. of the Computer Software and Applications Conference, vol. 2, pp. 41-42, 2014.

[28] S. Y. Yerima, S. Sezer, G. McWilliams, I. Muttik, "A New Android Malware Detection Approach using Bayesian Classification", In Proc.of Advanced Information Networking and Applications, pp. 121-128, 2017.

[29] Q. Jerome, K. Allix, R. State, T. Engel, "Using Opcode-sequences to Detect Malicious Android Applications", In Proc. of the IEEE Int. Conf. on Communicaions, pp. 914-919, 2014.