Mobile Botnet Sentinel Using CNN (Deep Learning Approach)

[1]Mandar Dixit, [2]Suyash Pardeshi, [3]Mahek Adhaduk, [4]Prof. Kirti Randhe

[1,2,3]Students, [4]Assistant Professor
Department of Computer Engineering,
ISBM College of Engineering, SPPU, India.

Abstract: **Android, as the most widespread mobile applications, is increasingly becoming the target of malware. Malicious applications designed to turn mobile devices into bots that may become part of a larger botnet are becoming increasingly common, thus posing a greater risk. This requires the most efficient ways to get the botnet on the Android platform. Therefore, in this project, we are using an in-depth learning botnet for Android botnet detection based on Convolutional Neural Networks (CNN). Our proposed botnet detection system is used as a CNN-based model trained in 342 static application features to distinguish between botnet applications and standard applications.**

Keywords: **Botnet detection; Android Botnets; Deep learning; Convolutional; Neural Networks; Machine learning; Android Botnets**

1. INTRODUCTION:

Android is currently the most widespread mobile operating system in the world. Over the years the volume of malware for Android targeted computer continued to grow. This is because it is easier and more convenient for non-computer programmers to identify an operating system that is open source, more common, and does not compromise the installation of applications from any possible source. In fact, many families of malware are capable of infecting Android devices and converting them into malicious bots found in the wild. These Android bots can be part of a larger botnet that can be used to carry out various types of attacks such as Distributed Denial of Service (DDoS) attacks, spam production and distribution, cybercrime, cybercrime, identity theft or credit theft. card details, etc.

A botnet contains a number of connected devices controlled by a malicious user or group of users known as a botmaster (s). It also contains Command and Control (C&C) infrastructure that enables bots to receive commands, receive updates and send status information to malicious players. Since smartphones and other mobile devices are commonly used to connect to online services and are not normally turned off, they provide a rich source of botnet performance. Therefore, the term 'mobile botnet' refers to a group of compromised phones and other remote devices for botmasters using C&C channels.

Today, malicious botnet applications have become a major threat. Additionally, their increasing use of sophisticated avoidance techniques requires more effective visual cues. Therefore, in this paper we present an in-depth learning method that uses Convolutional Neural Networks (CNN) to access the Android botnet.

LITERATURE SURVEY:

Sr. no.	Publication Details	Author	Abstract	Research Gap Identified
1	DeDroid: A Mobile Botnet Detection Based on Static Analysis	Ahmad Karim; Rosli Salleh; Syed Adeel Ali Shah	Mobile botnet phenomenon is gaining popularity among malware writers in order to exploit vulnerabilities in smartphones. In particular, mobile botnets enable illegal access to a victim's smartphone and can compromise critical user data and launch a DDoS attack through Command and Control (C&C).	Accuracy Very less
2	Mobile Guard Demo: Network Based Malware Detection	Vikramajeet Khatri; Joerg Abendroth	The growing trend of data traffic in mobile networks brings new security threats such as malwares, botnets, premium SMS frauds etc, and these threats affect the network resources in terms of revenue as well as performance. Some end user devices are using antivirus and anti-malware clients for protection against malware attacks, but the malicious activity affects mobile network elements as well.	Network issue

3	Mobile botnet detection: Proof of concept	Zubaile Mahmabdullah; Madihah Mohd Saudi; Nor Badrul Anuar	Nowadays mobile devices such as smartphones had widely been used. People use smartphones not limited for phone calling or sending messages but also for web browsing, social networking and online banking transaction. To certain extend, all confidential information are kept in their smartphone. As a result, smartphones became as one of the cyber-criminal main target especially through an installation of mobile botnet. Eurograbber is an example of mobile botnet that being installed via infected mobile application without victim knowledge. It will pretense as mobile banking application software and steal financial transaction information from victim's smartphone. In 2012, Eurograbber had caused a total loss of USD 47 Million accumulatively all over the world. Based on the implications posed by this botnet, this is the urge where this research comes in.	Less Accuarcy
4	Longitudinal performance analysis of machine learning by Android malware detectors	Suleiman Y. Yerima; Sarmadullah Khan	This paper presents a longitudinal study of the performance of machine learning classifiers for Android malware detection. The study is undertaken using features extracted from Android applications first seen between 2012 and 2016. The aim is to investigate the extent of performance decay over time for various machine learning classifiers trained with static features extracted from date-labelled benign and malware application sets. Using date-labelled apps allows for true mimicking of zero-day testing, thus providing a more realistic view of performance than the conventional methods of evaluation that do not take date of appearance into account. In this study, all the investigated machine learning classifiers showed progressive diminishing performance when tested on sets of samples from a later time period.	Not Efficient
5	Android botnets on the rise: Trends and characteristics.	H. Pieterse and M. S. Olivier	Smartphones are the latest technology trend of the 21st century. Today's social expectation of always staying connected and the need for an increase in productivity are the reasons for the increase in smartphone usage. One of the leaders of the smartphone evolution is Google's Android Operating System (OS). The openness of the design and the ease of customizing are the aspects that are placing Android ahead of the other smartphone OSs. Such popularity has not only led to an increase in Android usage but also to the rise of Android malware. Although such malware is not having a significant impact on the popularity of Android smartphones, it is however creating new possibilities for threats. One such threat is the impact of botnets on Android smartphones. Recently, malware has surfaced that revealed specific characteristics relating to traditional botnet activities. Malware such as Geinimi, Pjapps, DroidDream, and RootSmart all display traditional botnet functionalities. These malicious applications show that Android botnets is a reality. From a security perspective it is important to understand the underlying structure of an Android botnet.	Less Efficient

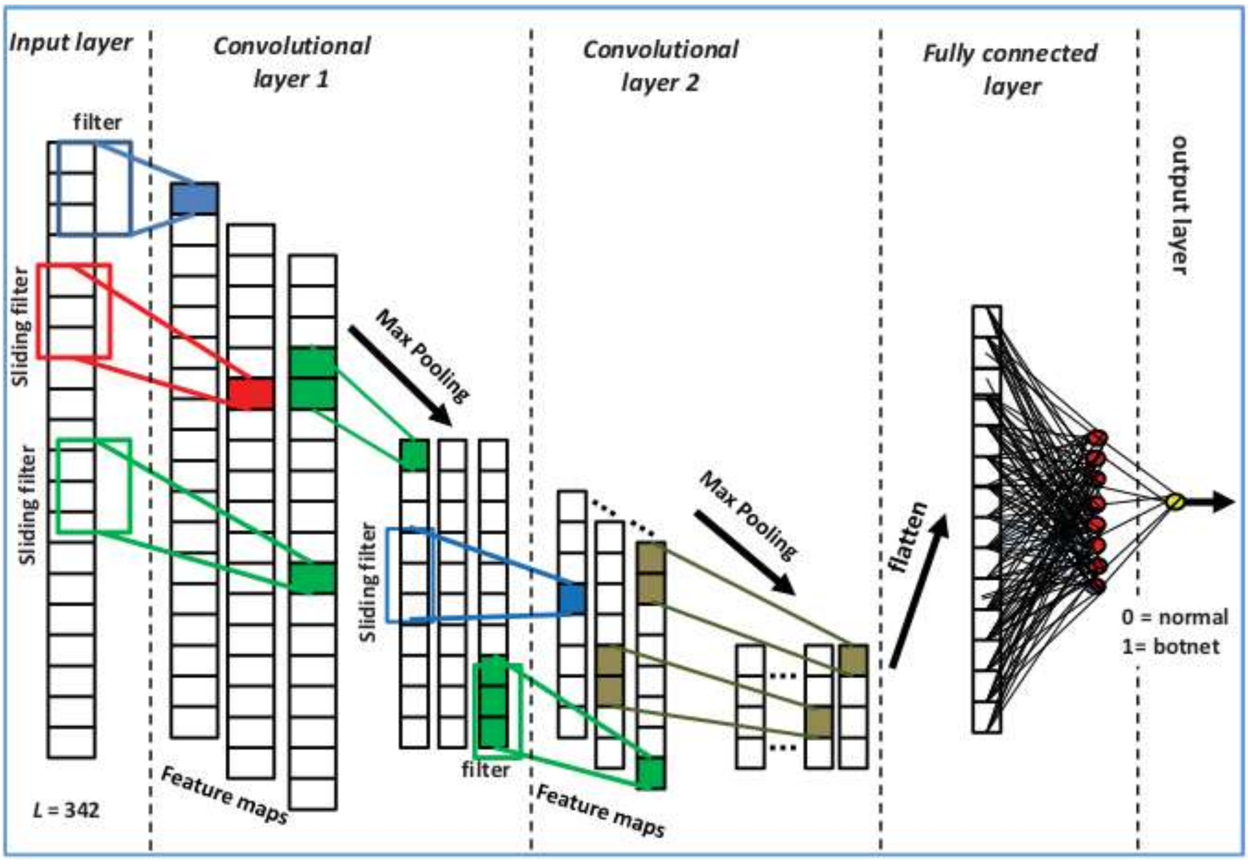
6	Toward a Detection Framework for Android Botnet	J. f. Alqatawna and H. Faris	Android is one of the most popular and widespread operating systems for smartphones. It has several millions of applications that are published at either official or unofficial stores. Botnet applications are kind of malware that can be published using these stores and downloaded by the victims on their smartphones. In this paper, we propose Android botnet detection method based a new set of discriminating features extracted based from the analysis of Android permissions (i.e. Protection levels for all available Android permissions). Then we compared the prediction power of different machine learning models before and after adding these features to the state-of-art requested permissions features in Android. We used four popular ML classifiers (i.e. Random Forest, MultiLayer Perceptron neural networks, Decision trees, and Naïve Bayes) for our experiments and we found that the new set of features have a tiny improvement on the performance in the case of decision trees and Random forest classifiers.	Method needs to be Improved
7	ABC: android botnet classification using feature selection and classification algorithms	Z. Abdullah, M. M. Saudi, and N. B. Anuar	Smartphones have become an important part of human lives, and this led to an increase number of smartphone users. However, this also attracts hackers to develop malicious applications especially Android botnet to steal the private information and causing financial losses. Due to the fast modifications in the technologies used by malicious application (app) developers, there is an urgent need for more advanced techniques for Android botnet detection. In this paper, a new approach for Android botnet classification based on features selection and classification algorithms is proposed. The proposed approach uses the permissions requested in the Android app as features, to differentiate between the Android botnet apps and benign apps.	Less Efficient

CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN is an in-depth learning method that is part of the Artificial Neural Networks family. It is effective in identifying simple patterns in data that will then be used to create complex patterns of higher layers. Two types of layers are commonly used to build CNNs; convolutional layers and integration layers. The role of the composite layer is to identify the local conjunctions of the elements from the previous layer, while the role of the composite layer is to merge the same elements in terms of logic into one. Generally, the convolutional layer produces the appropriate components while the compound layer reduces the magnitude of those features acquired in the convolutional layer (or - the previous composite layer). At the end of the model's tail, a layer (layers) that are fully connected (dense) are usually used to separate. Depending on the nature of the data, CNN performance may be influenced by the number of layers, the number of filters (kernels) or the size of the filters. In general, many invisible features are extracted from CNN's deeper layers, therefore, the number of layers required depends on the resolution and non-compliance of the analyzed data. In addition, the number of filters in each category determines the number of features removed. The complexity of the calculation increases with multiple layers and a higher number of filters. Also, with more complex structures, there is the possibility of training an over-installed model that leads to inaccurate predictor accuracy in test sets. To reduce overload, techniques such as 'dropout' and 'batch regularization' are used during the training of our models.

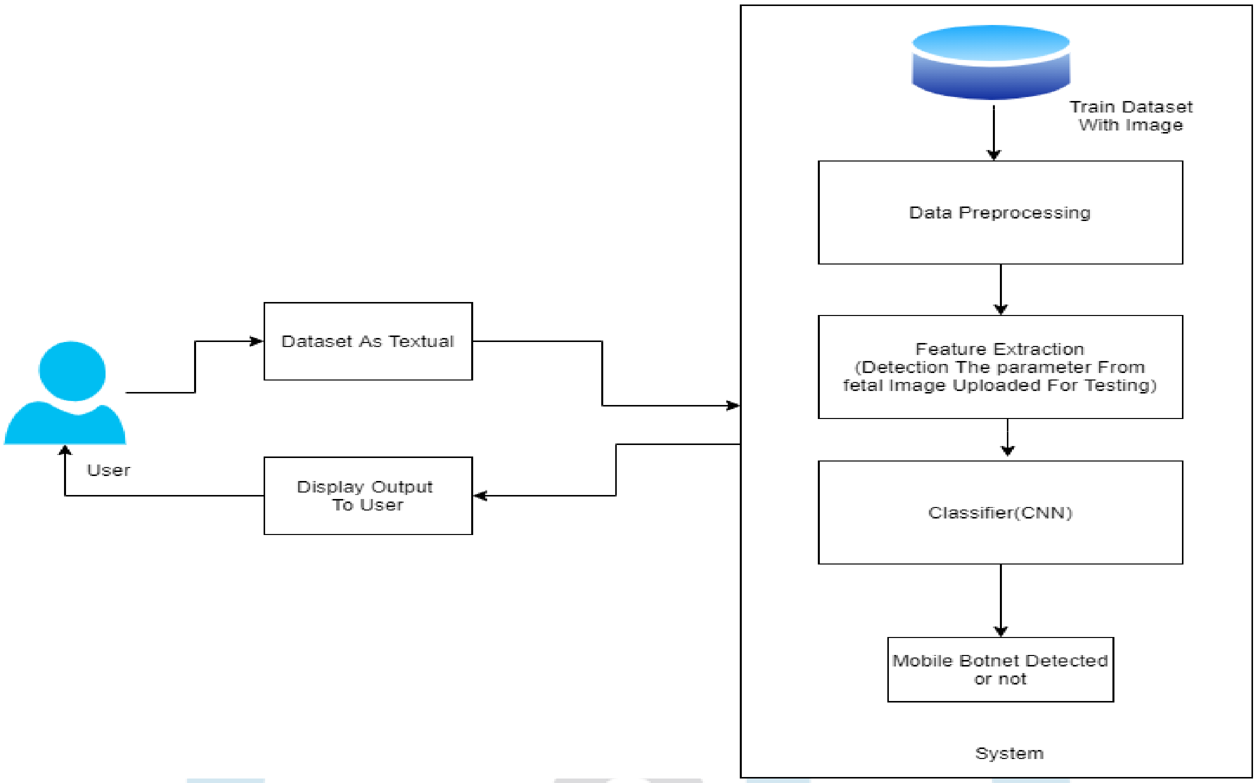
OUR PROPOSED CNN ARCHITECHTURE:

Our proposed CNN architecture is 1D CNN that combines two flexible layers and two large integration layers. This is followed by a fully interconnected layer of N units, which are also connected to the last dividing layer consisting of a single neuron with the function of sigmoid activation. The final layer of segmentation produces a result corresponding to two categories namely 'botnet' or 'normal'. The convolution layers use the ReLU (Rectified Linear Units) function provided by:. ReLU helps reduce the disappearance and explosion of gradient problems. It has been found to be more efficient in terms of time and cost of big data training compared to older non-linear opening functions such as Sigmoid or Tangent functions. A simplified view of our buildings is shown in Figure.



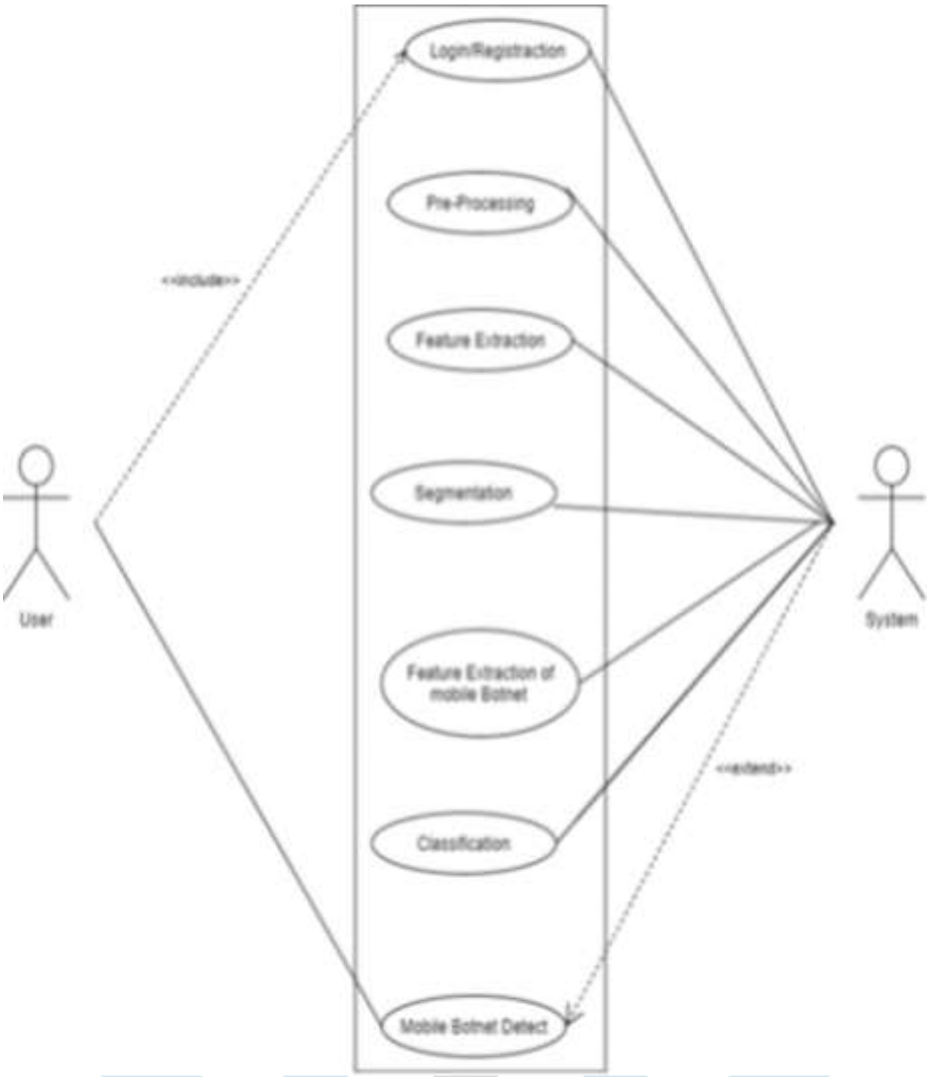
RELATED WORK:
In a study conducted by previous researchers, the aim was to address a gap in understanding mobile botnet and their communication features. Therefore, they have provided an in-depth analysis of Command and Control (C&C) and built-in URLs for Android botnet. By combining both static and dynamic and visual analysis, the relationship between the analyzed botnet families is revealed, providing insight into each malicious infrastructure. In this study a set of 1929 sample data sets for 14 Android botnet families was compiled and released to the research community. This paper and several previous activities on Android botnets used the full database or its own small set to test the proposed Android botnet detection strategies. They suggested a stand-alone access to the mobile botnet where they used the MD5 hash, permissions, broadcast receivers, and background services as features. These features have been removed from Android apps to create a machine learning component for mobile botnet attacks.

SYSTEM ARCHITECHTURE:
In the given structure after entering the application system will place the symbols. The ‘Pre-processing’ module will then delete the audio portion and unwanted data from the used database; we used a CSV file as a database. Pre-data processing is the process of preparing raw data and making it suitable for the Convolutional neural network model. It is not always the case that we get clean and formatted data. And while any data processing is done, it is compulsory to clean it and format it in a format. Therefore, in this case, we use the data processing function. In pre-processing data the data will become standard and converted to a single-vector binary binary format.


Train Dataset With Image
Data Preprocessing
Feature Extraction (Detection The parameter From fetal Image Uploaded For Testing)
Classifier(CNN)
Mobile Botnet Detected or not
System
Dataset As Textual
Display Output To User
User

USE CASE:

The diagram of the situation used in Integrated Model Language (UML) is a type of behavioral diagram that is defined and created from the Use-case analysis. Its purpose is to present an overview of the performance of the system provided by the characters, their terms (represented as terms of use), and any dependencies between those terms of use. The main purpose of the case diagram to use is to show what system functions are performed by which character. The roles of the characters in the program can be highlighted.



Use Case Diagram

CONCLUSION:

As smartphones become more popular, they become victims of potential attacks. With the openness of the Android OS design and its growing popularity, the growth of unprofessional Android software can be expected. In this paper specific trends and features of Android botnets have been identified. These can assist in the identification of current Android botnet and prevent the proliferation of new Android botnets. Future research includes an early study of the internal functionality of the current Android botnet and a malware program. The purpose of this study is to examine the development and basic structure of Android botnets to assist in the process of acquiring such botnet. A future focus will be on identifying Android botnet using a signature and / or behavioral-based acquisition model.

REFERENCES:

1. Del Rosso, Letteri, I., Del Rosso, M., Caianiello, P., Cassioli, D., 2018. Performance of botnet detection by neural networks in software-denied networks, in: CEUR WORKSHOP PROCEEDINGS, CEUR-WS.
2. Kadir, A.F.A., Stakhanova, N., Ghorbani, A.A., 2015. Android botnets: What urls are telling to us, in: International Conference on Network and System Security, Springer. pp. 78–91.
3. ISCX Android botnet dataset. Available from https://www.unb.ca/cic/datasets/android-botnet.html. [Accessed 03/03/2020]
4. S. Anwar, JM Zain, Z. Inayat, RU Haq, A. Karim, and AN Jabir, "A Stable Way to Obtain a Mobile Botnet," at the Third International Conference on Electricity Design 2016 (ICED), 2016: IEEE , pp. 563-567.
5. J. f. Alqatawna and H. Faris, "Before the Android Botnet Discovery Framework," at the 2017 International Conference on New Computer Science Trends (ITCCS), 2017: IEEE, pp. 197-202.
6. S Hojjatinia, S Hamzenejadi, H Mohseni, “Android Botnet Detection uses Convolutional Neural Networks” 28th Iranian Electronic Engineering Conference (ICEE2020).
7. Z. Abdullah, M. M. Saudi, and N. B. Anuar, "ABC: android botnet split using feature selection and classification," Advanced Science Letters, vol. 23, no. 5, pages 4717-4720, 2017.

8. Rosli & Shah, Karim, Ahmad & Salleh, Rosli & Shah, Syed. (2015). DeDroid: How To Get A Botnet For Mobile Based On Standby Analysis. 10.1109 / UIC-ATC-ScalCom-CBDCom-IoP.2015.240. Drebin Dataset. Available at: https://www.sec.cs.tu-bs.de/~danarp/drebin/index.html [accessed 05/03/2020]

9. Madhav, J., Dutt, S., Calangutkar, K., Oh, T., Kim, YH, Kim, JN, 2015. Cloud based android malware discover system, in: Advanced Communication Technology (ICACT), 2015 17th International Conference kwe, IEE. pages 347-352.

10. S. Y. Yerima, M. K. Alzaylaee, and S. Sezer. "A flexible analysis based on machine learning for Android applications with advanced coding" EURASIP Journal on Information Security, 4 (2019). https://doi.org/10.1186/s13635-019-0087-1

11. Meng, X. and Spanoudakis, G. (2016). MBotCS: Computer-based botnet detection system, 9572, pages 274-291. doi: 10.1007 / 978-3-319-31811-0_17

12. B. Alothman and P. Rattadilok 'Android Botnet Discovery: The Source of Source Code' 12th International Conference on Internet Technology and Secure Transactions (ICITST), 11-14 Dec., Cambridge, UK, 2017, IEEE, pp 111-115.

13. A. J. Alzahrani and A. A. Ghorbani, "How to get real-time signature sms botnet," at the 2015 Thirteenth Annual Conference on Privacy, Security and Trust (PST), 2015: IEEE, pp. 157-164.

14. D. A. Girei, M. A. Shah, and M. B. Shahid, "Improved botnet mobile device optimization strategy using log analysis," 2016's 22nd International Conference on Automation and Computing (ICAC), 2016: IEEE, pp. 450-455.

15. M. Yusof, M. M. Saudi, and F. Ridzuan, "A New Android Botnet Classification of GPS Exploitation Based on Permission and API Calls etc," at the International Conference on Advanced Engineering Theory and Applications, 2017: Springer, pp. 27-37.

16. Y. LeCun, Y.Bengio, and G. Hinton, In-depth Reading, Nature 521 (2015), no. 7553, 436-444

17. N. Srivastava, G. Hinton, A. Krizhevsky, I. Stuskever, noR. Salakhutdinov. "Discontinuation: A method of preventing neural networks from overloading" Journal of Machine Learning Research, 15 (1): 1929-1958, 2014.