FNN and Auto Encoder Deep Learning-Based Algorithm for Android Cyber Security

Seema Vanjire, M. Lakshmi

***Abstract:** Android is susceptible to malware attacks due to its open architecture, large user base and access to its code. Mobile or android malware attacks are increasing from last year. These are common threats for every internet-accessible device. From Researchers Point of view 50% increase in cyber-attacks targeting Android Mobile phones since last year. Malware attackers increasingly turning their attention to attacking smartphones with credential-theft, surveillance, and malicious advertising.*

Security investigation in the android mobile system has relied on analysis for malware or threat detection using binary samples or system calls with behavior profile for malicious applications is generated and then analyzed. The resulting report is then used to detect android application malware or threats using manual features.

To dispose of malicious applications in the mobile device, we propose an Android malware detection system using deep learning techniques which gives security for mobile or android. FNN(Fully-connected FeedForward Deep Neural Networks) and AutoEncoder algorithm from deep learning provide Extensive experiments on a real-world dataset that reaches to an accuracy of 95 %. These papers explain Deep learning FNN(Fully-connected FeedForward Deep Neural Networks) and AutoEncoder approach for android malware detection.

***Keywords:** Auto Encoder, Android Cyber Security, Deep Learning.*

I. INTRODUCTION

Mobile is a handheld device made for portability, an attribute owing to its compactness and light-weight. New data storage, processing, and display technologies have been allowed to do anything on mobile devices. Mobile application purpose for delivering data, services, and ease to users using IT strategies that are based on securing the network perimeter of the application. When we talk about mobile app security, the app developer needs to deliver critical endpoint security. Online users using mobile apps that provide a better user experience or streamline operations are more exposed to a range of threats.

For customer-facing applications, businesses have trusted their security to coding practices, number of internal security and app penetration tests. These capabilities build layers of defense hence providing the most effective security. application protection solutions have been used by businesses to protect mobile applications, they only protect against what is known at the time they are deployed.

Most current app security solutions don’t possess the ability to understand how protections are holding up in the wild, only providing insight into new threats that can’t be countered in real-time.

Protecting consumer-facing mobile apps requires:

- The ability to make app code extremely difficult to reverse engineer
- Ensure critical data and key security, and
- Report real-time app threat status.

Enabling these requires building three defensive layers to adequately address the emerging threat landscape:

1. Application protection using a configurable guard network to obscure code and harden the application and to enable Runtime Application Self-Protection Security (RASP), tamper resistance and self-healing measures.

2. White-Box Cryptography to encrypt and protect critical communication key and data

3. Real-time threat analytics to provide an understanding of the threat posture of apps

Android Security report states that the ‘Play Protect’ software scans over 50 billion apps on a daily basis across more than 1.6 billion devices. According to the report, Google Play Protect protected over 1.3 billion devices from installing PHAs from outside sources in 2018. Coming to the percentage of devices that installed PHAs- Android OS version wise shown in table 1. Also, Fig 1.shows Pie chart for Table 1.

Table 1. Pha percentage for android os device

Android Device OS Version	*PHA Percentage*
Lollipop-	0.65%
Marshmallow-	0.55%
Nougat-	0.29%
Oreo-	0.19%
Pie-	0.18%

Manuscript published on January 30, 2020.
* Correspondence Author

Seema Vanjire*, Assistant Professor, Sinhgad Academy of Engineering, Kondhwa Pune, India.

Dr. M. Lakshmi, Professor, Department of CSE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Kuthambakkam, Tamil Nadu, India.

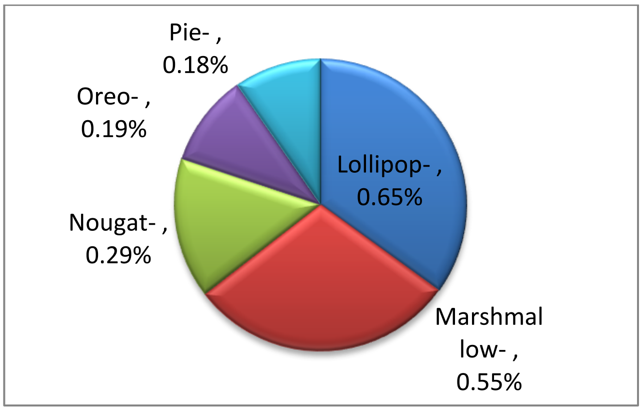

Fig1. Pie Chart for Table 1

One of the problems faced in Android security is that it annually pays million through rewards program to offer encouragement to the security researchers from around the world to improve Android Security. As per cybersecurity or android security researchers, 25 million Android mobile devices have been hits with malware now being called 'Agent Smith.' The malware hides within apps like WhatsApp or Facebook, taking advantage of the vulnerabilities within the Android operating system. These new malware apps available on the play store and injected with its malicious code – replacing the original application with viral version. The attacked apps work just fine on the mobile platform but the malware is hidden from real users. The malware then displays unwanted advertisements to users, which may not seem like a big problem at the start but the same security flaws could be used to hijack net banking, online shopping, and other sensitive personal information.

II. ANDROID SECURITY

Mobile malware attacks are rising in last year. These are attacking smartphones with credential-theft, surveillance, and malicious advertising.

- The ransomware crisis is going to get a lot worse.
- LastPass bug leaks credentials from the previous site.
- Database leaks data on most of Ecuador's citizens, including 6.7 million children.

This suggests one of the key reasons for the sharp rise is the increased use of mobile banking applications. The sharp rise in mobile banking malware correlates to the growing use of mobile banking applications, Maya Horowitz, director of threat intelligence and research at Check Point, told ZDNet. Some forms of Android malware and threats developed with advanced elusion techniques in order to remain undetected on infected devices. For example, banking Trojan will start operating after motion sensors detect that the device has been moved a strategy to avoid it being detected and analyzed in sandbox environments.

Nowadays for every android device, users keep the internet on or some applications such as Whatsapp, Facebook, Twitter or any news applications running in the background. these applications are continuously accessing battery and internet. As they use internet data leakage data loss may occur at many attempts. And this is time for attackers to gain access to an android system for an attack.

Attackers do an attack on these applications Whatsapp, Facebook, Twitter or any news applications at any moment and destroy the application data and mobile sensor data. This is rising day by day. Many times it is observed that these attacks are following the same name as an alike android file or attacks stay hidden inside any other application. So it is difficult to maintain security for the mobile handheld device. Thus this paper proposes new android detection system using deep learning algorithm such as a supervised learning-based FNN algorithm and unsupervised based autoencoder algorithm

III. DEEP LEARNING APPROACH FOR ANDROID SECURITY

To contrive with malicious applications that are increased in volume and sophistication, we propose an Android malware detection system using deep learning techniques to face the threats of Android malware.

All Deep learning algorithms are based on Deep Neural Networks (DNN), which are organized in many layers capable of autonomous representation learning. Table 2 shows different Deep learning algorithms used for malware detection. This table also shows up to now the not any supervised algorithm is used for spam detection or phishing detection.

Table 2. Deep Learning Algorithm Classification for malware Detection

Algorithm Type	Malware Detection	Spam/Phishing Detection
Supervised learning	Fully-connected FeedForward Deep Neural Networks (FNN)	——
	Convolutional FeedForward Deep Neural Networks (CNN)	——
	Recurrent Deep Neural Networks (RNN)	——
Unsupervised learning	Deep Belief Networks (DBN	Deep Belief Networks (DBN
	Stacked Autoencoders (SAE).	Stacked Autoencoders (SAE).

A. Supervised DL Algorithm For Malware Detection

Supervised DL algorithms For malware detection are FNN(Fully-connected FeedForward Deep Neural Networks), CNN(Convolutional FeedForward Deep Neural Networks), RNN(Recurrent Deep Neural Networks).

In FNN every neuron is connected to all other neurons which are present in the previous layer and next layer. FNN no make any supposition on the input data. FNN provides a flexible good solution for classification problems.

Next, in CNN each neuron takes input from previous layers nodes or neurons. This is more effective in spatial data analysis. But performance degrades for nonspatial data.

Next in RNN, Each neuron sends output to previous layer neurons. This makes the design complicated. And also more costly from the development view.

B. Unsupervised DL Algorithms For Malware Detection

Here, For malware detection, phishing and spam detection DBN(Deep Belief Networks) and SAE(Stacked AutoEncoders) algorithms are used.

This DBN is one standard class of neural networks with no output layer.

This is mostly and fruitfully used for pretraining tasks for example feature extraction. Next in SAE neural network is made up of multiple autoencoders where the number of inputs is equal to the number of outputs taken from the same number of nodes and given to the same number of output nodes. This attain better result on small datasets for unsupervised DL algorithm

IV. SYSTEM DESIGN

Various solutions have been proposed for malware detection which includes static and dynamic methods. However, these solutions are platform-specific or can perform better only in some particular execution environment. The intention here in the context of the malware detection and thereat attribution is to provide the portable framework for malware detection across all the platform and execution environments.

The security investigators can rely on the plug and play framework.

The Feature Extraction reports are analyzed with classifiers to detect malware and threats.

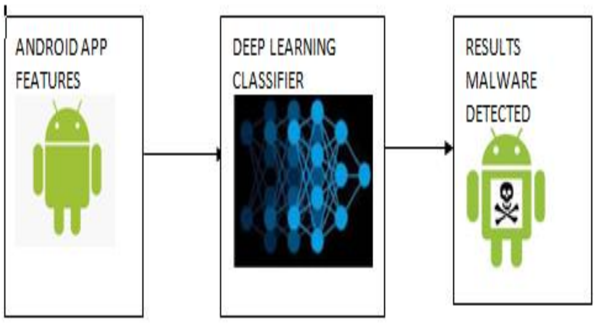

Fig 2. System Design

In system design as in above Fig 2., Android Application features are first extracted then different Deep learning algorithm is applied on features Dataset then after analysis the result will be shown for malware or threats are detected.

In the above system design, there are Three major working parts

1. Collection And Training Dataset
2. Extraction Of Features
3. Deployment Of Deep Learning Algorithm

In the below section working of the above Fig is explained by considering the above parts.

A. Collection and Training Dataset

The first part is the collection of datasets. a large collection of malware apps have been collected to form a large dataset. The generated dataset is used for training the model with given parameters. Data preprocessing is an important part of this process where the data must be organized.

B. Extraction Of Features

The feature extraction is important for prediction and selected meticulously in order to perform. With the collected malicious app the features are taken from source codes of decompiled files. the installation packages of Android apps are basically .apk file that can be decompiled for dataset to train the classifier

C. Deployment of Deep Learning Algorithm

Now after obtaining features, the deep learning algorithms are used for classification. during the training process, a set of labels is set to determine the type of each application. Here we are using FNN and AutoEncoder deep learning-based approach for classifying dataset. Which gives a result with near about 95 % accuracy

V. METHODOLOGY

The system is designed as per a security analyst point of view. The first step is analysts login .at A successful login application provides a page for a selection of different analysis functionality such as feature extraction for the android app. Behavior reports are then classified using a deep learning algorithm. The system framework implements various Deep learning-based algorithms for malware detection such as FNN and AutoEncoder. These algorithms are explained in the below section in detail.

From the above Deep learning algorithms, FNN and AutoEncoder approaches are Least used for malware detection but the survey says in other applications it approaches to near about 95% to 98% accuracy.

A. FNN Deep Learning Algorithm

FNN is a Deep FeedForward network with multiple layers. The aim of this network is to estimate major functions in the neural network. In this network, multiple connections between multiple layers do not form a cycle.

They are called FeedForward network because data and information only travel in the forward direction, first through the input layer nodes, then through the hidden layer nodes, and then finally through the output layer nodes to end nodes.

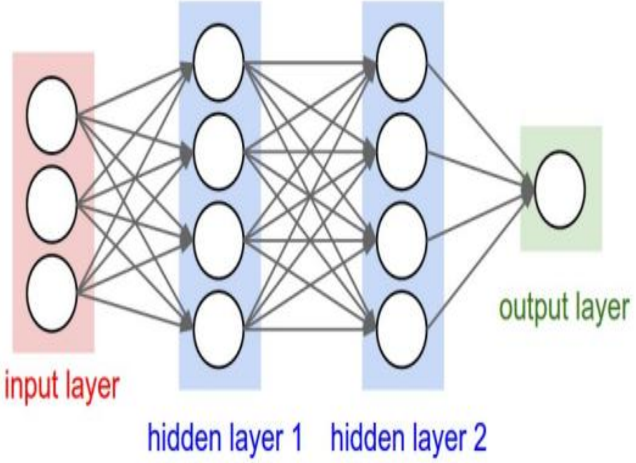

Fig 3. Four-layer FeedForward neural network

The above Fig. 3 shows the structure of a four-layer FeedForward neural network which contains an input layer output layer and two hidden layers. The number of hidden layers may increase as per the structure of the neural network. In the above Fig 3. this is clear since each layer is only connected to the layer directly to its left. FeedForward neural networks have a greatly simplified learning algorithm.

B. AutoEncoder Neural network

An AutoEncoder is one of the good neural networks for an unsupervised learning approach .this This has three layers: an input layer, a hidden layer or code, and an output layer.

This network is trained to reform its inputs, which services the hidden layer to learn the high-quality representation of the inputs.

An AutoEncoder neural network is an unsupervised deep learning algorithm that applies backpropagation, to the inputs. An AutoEncoder is trained to provide input to its output. Internally, it has a hidden layer that describes a code which is further used to represent the input.

AutoEncoders, are more flexible and effective to represent both liners and non-linear transformation in the encoding and decoding side of the network. These can be layered to form a deep learning network due to its Network representation and structure.

C. Sparse AutoEncoder

This is a simple single-layer sparse AutoEncoder. The hidden nodes are activated. The activation of this depends on the input.

1) Recently, it has been observed that when representations are learned in a way that encourages sparsity, improved performance is obtained on classification tasks.

In Fig. 4, the sparse AutoEncoder structure is elaborated with three layers as the input layer, the code layer, output layer. The encoder works together with the input layer and Code layer. Similarly, Decoder works together with the Code layer and Output layer. Sparse AutoEncoder with hidden layer is shown in Fig. 5 below

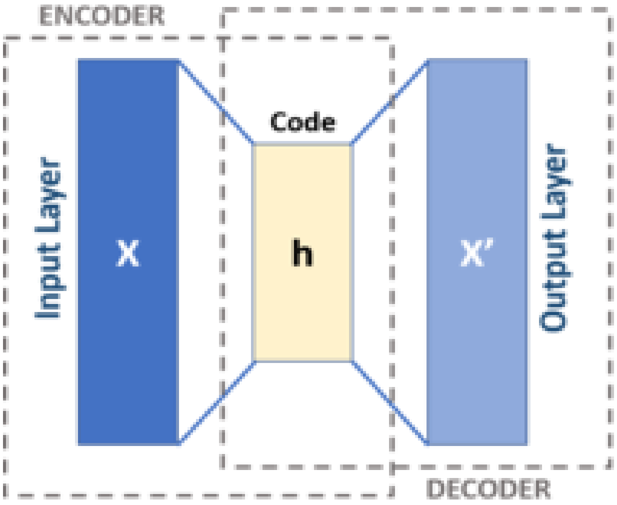

Fig 4. Sparse AutoEncoder.

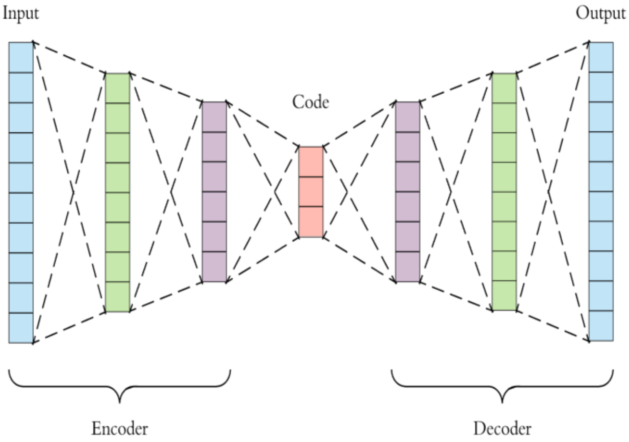

Fig 5. Sparse AutoEncoder with hidden layers

Here in Fig 5, At the Encoder side, the number of hidden layers present in between the input layer and code layer similarly at the decoder side also the number of hidden layers present in the code layer and output layer.

VI. RESULT ANALYSIS

After training of dataset and finding permission-based features, a model is prepared. A relatively unknown application is sent as new data for predicting malicious apps .thus Highly sensitive apps are used as final features of the new app. These features are then employed as input to obtain the result of the app as malware or not.

Using these algorithms accuracy calculated for both algorithms and shown in below table

Table 3. malware Detection Rate

Deep Learning Algorithm	Malware Detection Rate Of Android App
FNN	96%
Autoencoder	98%

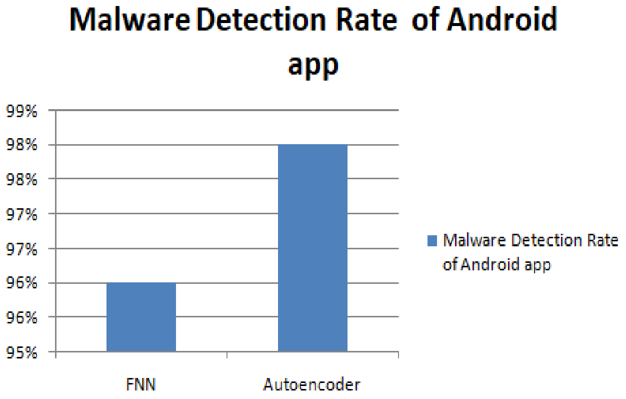

Fig 6. Malware Detection Rate

In above figure 6, It shows the resultant graph for Malware Detection Rate using the FNN algorithm is 96% and the malware Detection Rate using the Autoencoder algorithm is 98%.

VII. CONCLUSION

Every day, the number of malware generated and targets the well-being cyberspace which is increasing exponentially. it leads to overwhelms the security investigators. Feature extracted reports are first generated and further analyzed using deep learning algorithms to produce the result for malware detection. In this paper, we propose a portable effective and efficient framework for malware detection using advanced FNN and AutoEncoder deep learning-based approaches. This will result in near about 95% o 98% accuracy for malware detection.

REFERENCES

1. Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, "On the Effectiveness of Machine and Deep Learning for Cyber Security", International Conference on Cyber Conflict, 2018
2. Mohamed Loey, Ahmed El-Sawy, Hazem EL-Bakry, "Deep Learning AutoEncoder Approach for Handwritten Arabic Digits Recognition",2015.

3. Padmavathi Ganapathi, Shanmugapriya, “Handbook of research on the machine and Deep learning Applications for Cyber Security ”, IGI Global Publication, DOI:-10.4018/978-1-5225-9611-0, July 2019
4. TT Teoh, Graeme Chiew, Yeaz Jaddoo, H.Michel,” Applying RNN and J48 Deep Learning in Android Cyber Security Space for Threat analysis”, DOI: 10.1109/ICSCEE.2018.8538405, 11-12 July 2018
5. Roman Graf, L. Aaron Kaplan, Ross King, “Neural Network-Based Technique for Android Smartphone Applications Classification”, DOI: 10.23919/CYCON.2019.8757162, 28 May 2018
6. Abdelmonim Naway, Yuancheng Li, “A Review on The Use of Deep Learning in Android Malware Detection”, May 2018
7. Krzysztof J. Geras and Charles Sutton, “Composite denoising AutoEncoders” 2017
8. Fu-Qiang Chen, Yan Wu, Guo-dong Zhao, Jun-ming Zhang, Ming Zhu, Jing Bai, “Contractive De-noising Auto-encoder”, yanwu@tongji.edu.cn
9. Qinxue Meng∗ , Daniel Catchpoole†, David Skillicorn‡, and Paul J. Kennedy, “Relational AutoEncoder for Feature Extraction” CS.LG 9 Feb 2018
10. Jianlin Xu, Yifan Yu, Zhen Chen, Bin Cao, Wenyu Dong, Yu Guo, and Junwei Cao,” MobSafe: Cloud Computing Based Forensic Analysis for Massive Mobile Applications Using Data Mining” TSINGHUASCIENCEANDTECHNOLOGY ISSNll1007-0214ll10/10llpp418-427 Volume 18, Number 4, August 2013
11. Suleiman Verma, Sakir Sezer, “Android Malware Detection Using Parallel Machine Learning Classifiers “, 8th International Conference on Next Generation Mobile Applications, Services and Technologies, (NGMAST 2014), 10-14 Sept. 2014
12. Lokesh Vaishanav1, Shanu Chauhan1, Hrithik Vaishnav, “Behavioural Analysis of Android Malware using Machine Learning “, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 6 Issue 5 May 2017, Page No. 21378-21389
13. Sindhu. K. K., Dr. B. B. Meshram “A Digital Forensic Tool for Cyber Crime Data mining “, IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Vol.2, No.1, 2012
14. Ashwinkumar Malwadkar, Prof. Sonali Patil “Data mining Techniques for Digital Forensic Analysis”, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 4 Issue: 3

AUTHORS PROFILE

Seema Vanjire is a Research scholar at Sathyabama Institute of Science and Technology Chennai. She has completed MTech CSE IT From Vishwakarma Institutes and Technology, Pune. Seema is working as Assistant Professor in Sinhgad Academy of Engineering , Kondhwa Pune. Currently, her research area is Cyber Security, Machine Learning , Data Mining And Deep Learning..

Dr. M Lakshmi, presently working as Professor in Department of CSE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences. She previously worked as Principal in Sri Krishna College of Technology, Coimbatore from Sep 2018 to April 2019. She also worked as Professor and Dean, School of Computing, Sathyabama University, Chennai, till June 2018 from Nov 1995. She has an experience of more than 24 years in Teaching. She has completed her B.E. (Computer Science and Engineering) from Bharathidasan University and M.E. (Computer Science and Engineering) from Madras University and Ph.D. from Sathyabama University in the area of Wireless Ad Hoc networks.