# A Deep Learning Approach to Malware Detection in Android Platform

**Abdulrazak Yahya Saleh, Corrine Francis**

***Abstract*: Throughout the years, mobile devices such as tablets, smartphones and computers are extremely widespread because of the development of modern technology. By using these devices, users all over the globe can easily access a huge range of applications from both commercial and private use. Malware detection is an important aspect of software protection. As a matter of fact, the development of malware had begun soaring as more and more unknown malware are discovered. Malware is a common term used to describe malicious software that can induce security threats to any device and also to the Internet network. In this paper, malware detection based Deep Learning approach utilizing the Long-Short Term Memory Networks (LSTM) algorithm is conducted by the researchers. The chosen approach learns and trains itself using the features that are needed for malware detection. Then, large data sets are used for evaluating the trained algorithm.***

***Index Terms*: Android Platform, Deep Learning, Long-Short Term Memory, Malware, Malware Detection.**

## I. INTRODUCTION

According to [1], Android malware is uncommon but they occur. Android malware were created to quietly command a gadget, embezzle credential data or money from the gadget's holder, and steal security passwords together with bank account numbers [2]. Portable gadgets such as tablets and smartphones have spread via modern civilization at an exponential tempo and as a matter of fact the usage of the gadgets have blossom for a while. According to [3], the quantity of movable gadgets in use will expand from over 11 billion within 2016 to over 5 billion addition by 2020. Within the gadget platforms, Android has become one of the most notable Gaia of all the gadget platforms. In hock to the open-door nature of applications (apps) amelioration as against to the "confined oasis" kind of workflow tailed by its rivals, it has achieved idolization at a tremendous stride. Since there is much crucial private information being kept on these computing gadgets, it is easy for the malware creators and hackers to gain that information. It is because Android is an open fountainhead with ground-level barricades to break in and thus may avail to become the target for malware invasion. Malware is malevolent software that acts to creep into the concealment and integrity of a system [4] such as, in [5] it guise grave dangers to the smartphone users. The malware will steal classified data, and also can dispatching memo and ad without the user permission [6]. As stated by the mobile warning report liberated by F-Secure in 2014 [7], estimated 95% of bad-natured apps were released on the Android platform. All the malware creators have turned their concentration from Windows of the PC generation to Android of the gadget aeon [8- 9].

Based on a study conducted in 2015 that advised over 6.3 million apps, nearly 1 million were discover to be malware dropping in greater than 250 classifications as stated by Symantec [9]. Addition to that, in 2015 there are nearly 750,000 fresh Android malware models were discovered, a 32% thrive from 2014 [9] and this flow is anticipated to get atrocious. Furthermore, according to [10], the Android gadgets are being contemplated by a huge part of malware.

The threat that malware hold is bringing a negative effects in both emotional and financial as stated by a newest report by Kaspersky Lab [11] that there are almost 1 billion dollars that being stole within approximate two years from the entire monetary establishment all around the globe as the aftereffect of malware invasions. Because of that, the immediate obligation for capable safeguard techniques to shield all the Android-enabled gadgets that have the functions of identification and prevention with the malware are needed.

In January 2017, the detection of malware infection was at 43% and since then there was a soaring activity of malware infection where there is approximately 56% of new and unknown malware being detected in April 2017 [12]. Smartphones possessed a huge pile of classified information that includes financial security and personal details. That quantity of valuable information can be retrieved illegally from the smartphones had made it the main mark for cyber law-breaker from all stripes [13]. Therefore, the malware detection is important as it can prevent illegal hackers from stealing user's credential information [13]. Deep Learning (DL) is the software that tried to imitate the activity in layers of neurons in the human brain, specifically in the neocortex segment that consists of 80% wrinkly part where the process of thinking happens [14]. DL's main function is to learn how to represent data [15]. There are numerous researches that incorporate the utilization of DL technique in the malware detection for Android platform [16-18]. The method of using DL in malware detection has inspired a large quantity of triumphant programs in communication identification, figure categorization, and natural language concoct. An introductory chore in DL as it administer to Android malware recognition was bestowed as explained in [18]. However, there are very restricted concentration has been reimbursed to

**Revised Manuscript Received on June 05, 2019**

**Abdulrazak Yahya Saleh**, FSKPM Faculty, University Malaysia Sarawak (UNIMAS), Kota Samarahan, 94300 Sarawak, Malaysia.

**Corrine Francis**, FSKPM Faculty, University Malaysia Sarawak (UNIMAS), Kota Samarahan, 94300 Sarawak, Malaysia.

employ the deep layouts to the aegis jurisdiction. This is because most of the DL prototypes are special designs demanding an aspiration to befittingly create them regardless of the ubiquity of attainable and blatantly applicable instruments [19].

The Long-Short Term Memory (LSTM) is one of the examples of DL where it has the capability of learning long-term interdependence [20]. Furthermore, it also had been widely used in the language modelling and proven to perform well [21-22]. The utilization of LSTM in Android malware detection was still fresh as it started to arise in 2017 [23-24].

## II. METHODS

This section reviews the vital foundation of Long-Short Term Memory (LSTM) and discusses the algorithm for this Deep Learning technique. In the first part, an introduction of LSTM has been presented. Finally, the second part focuses on the proposed LSTM algorithm which is used to improve and enhance the performance of malware detection.

### A. Long-Short Term Memory (LSTM)

LSTM technique is proposed by Hochreiter and Schmidhuber in 1997. It is a feed-forward network with one or more hidden layers where its main purpose is to learn long-term reliance [20]. LSTM possessed the framework like chain and the repeating module contains dissimilar arrangement. There are four neural network layers that interact in a special way with each other. Gibson and Patterson [25] stated that each LSTM unit contains two classes of connections which are the connections from the preceding time-pace (unit outputs) and the connections from the earlier layer. The crucial parts of the LSTM architecture is the memory cell and the gates (also includes the forget cell and the input cell). The materials of the memory cell are adjusted by the input gates and forget gates. Presume that both of the gates are closed, then the materials inside the memory cell will stay unaltered between one time-paces and the next. In summary, the input gate defends the unit from unnecessary input incident, the forget gate helps the unit to erase the past memory materials and the output gate unveil the materials of the memory cell at the LSTM output unit.

For more visualization of the recurrent connection on the hidden layer nodes, Figure 1 shows the process.

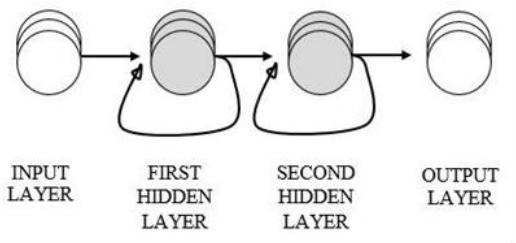


**Fig. 1**: Recurrent connection in the hidden layer nodes

Then, the further visualization demonstrates how the input signals "slides" through the overall network using the feedforward and across time in Figure 1 has been shown in Figure 2.

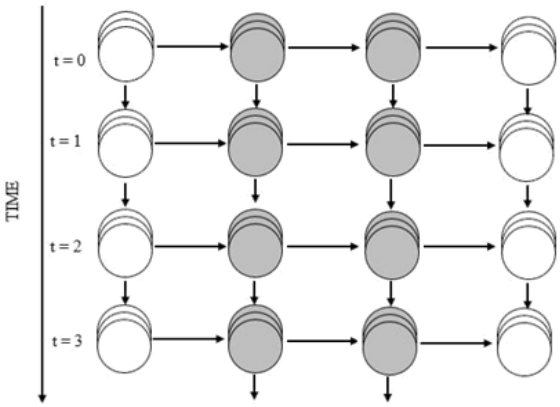


**Fig. 2**: Input signals framework moving through the hidden layers

The main idea that permits the LSTM network to conserve condition over time is the memory cell [25] or also known as the cell state [20] the horizontal line in the LSTM network. The cell state, as shown in Figure 3, runs in the whole net-work only with a little linear interaction thus making the information easier to slide through it unaltered.

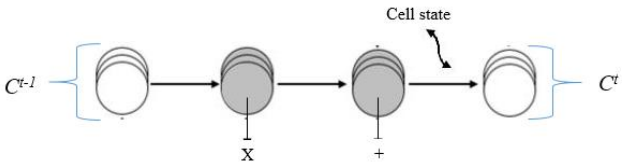


**Fig. 3**: Memory cell or cell state

Gates, as in Figure 4, in the other hand, is the way to deliberately let information pass. They are elements of a sigmoid neural net layer and a pointwise duplication activity. 0 and 1 will be the sigmoid output numbers, explaining how much of a single segment should be pass. 0 value means "nothing pass" and 1 means "everything pass".

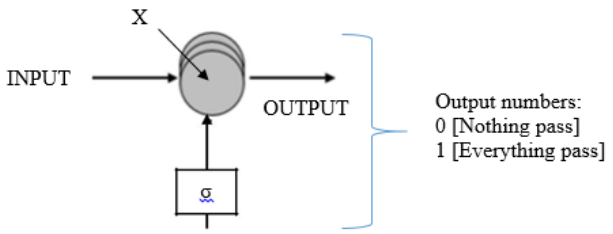


**Fig. 4**: Sigmoid layer

The principal structure of the LSTM is accredited as the LSTM block as shown in Figure 5.

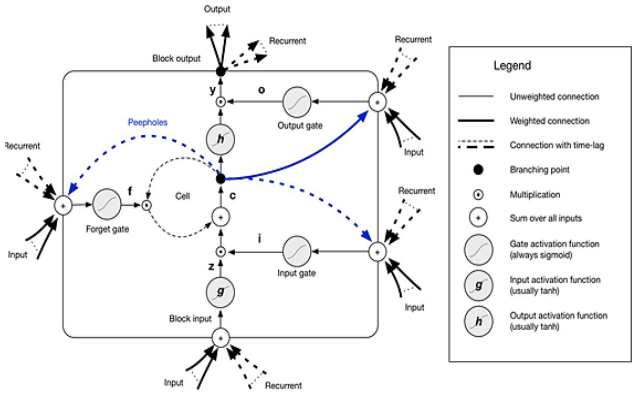


**Fig. 5**: LSTM block illustration [25]

### B. LSTM for Android Malware Detection

This section discusses the proposed LSTM algorithm that has been used in the malware detection technique. This algorithm determines the malware detection structure and its corresponding model parameters. For the general framework of this study, there are 5 levels where it is shown in the Figure 6 below:

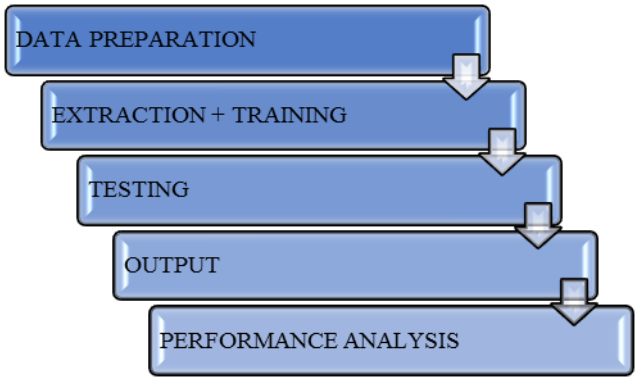


**Fig. 6**: General Framework

1) *Data Preparation*: The dataset used within this paper is gathered from the Drebin Dataset where it contains over 179 dissimilar malware groups, 138047 malicious samples and 54 features. Approximately 80% of the dataset is categorized as malware. The dataset were retrieved via online in the GitHub website [26] where the final dataset consists of 96724 malicious and 41323 benign data. This dataset were created to aid in the Android malware detection research [27].

2) *Extraction + Training*: The samples inside the Drebin dataset are processed through the LSTM algorithm for features extraction. The samples have been used as the information for the training process. For the training process, the information that extracted from the dataset has been chosen randomly.

3) *Testing*: In this process, after the information being selected randomly the LSTM algorithm has been used for the training then the validation process is performed after the training process finish.

4) *Output*: The final output is display using the line graph by comparing both of the training and validation accuracy.

5) *Performance Analysis*: The output of the algorithm is measured using the accuracy metric where it displays an output from the training accuracy and also from the validation accuracy.

## III. RESULTS AND DISCUSSIONS

This section displays the results of the proposed method LSTM based on three types of experiments: 1) Comparison of the dataset size, 2) Accuracy, and 3) Compared to other method which is the Back-Propagation. The results that are achieved by these experiments are measured in terms of accuracy. The experiments are run on numerous times in the training and testing for the dataset.

### A. Dataset Sizes

The large samples are divided into two small samples as shown in Figure 7 below.


**Fig. 7**: Different samples for training experiment

Then, the algorithm is trained using the different samples together with the selected hyper-parameter as shown in Table1 below.

**Table 1**: The best hyper-parameter configuration on the training set

| Hyper-parameter | Best configuration |
|---|---|
| Batch size | 64 |
| Epochs | 130 |
| Activation | Sigmoid |
| Optimization | Adam |

Experiments have been conducted using the two different sizes of samples. Based on the results, as shown in Table 2, it can be observed that the accuracy metric for the sample larger than 5, 000 accuracy did not match with the accuracy training line. Apart of that, when conducting this experiment, the time taken for both of the large samples (50, 000) to complete its training and validation were slow and takes a long duration of time. The accuracy achieved by the large samples is constantly resulted in lower accuracy below 30%. Based on the result, the utilization of the 5000 samples from the Drebin dataset is used for the next experiment.

**Table 2**: Accuracy comparisons for different samples

| Sample | Accuracy |
|---|---|
| 5, 000 | 76.48% |
| 50, 000 | 27.20% |

### B. Accuracy Performance

In this experiment, 5000 data samples were extracted from the Drebin dataset and some features from the 54 features will be selected randomly according to its importance for the malware detection. The hyper-parameter used for the LSTM algorithm is shown in Table 1. After the selected hyper-parameter was tuned into the LSTM algorithm, the processes of training and validation were conducted in

order to achieve the accuracy. As a result, based on Figure 8, it only takes less time to process all the extracted features from the sample as the sample size were small and an accuracy of 93.60% was successfully achieved. Figure 9 shows the statistical graph of comparison between the model loss and accuracy based on the training and validation processes. It can be observed that both of the model loss (loss: 0.6334, val_loss: 0.6333) and model accuracy (acc: 0.9446, val_acc: 0.9360) nearly achieved the same results for the training and validation steps.

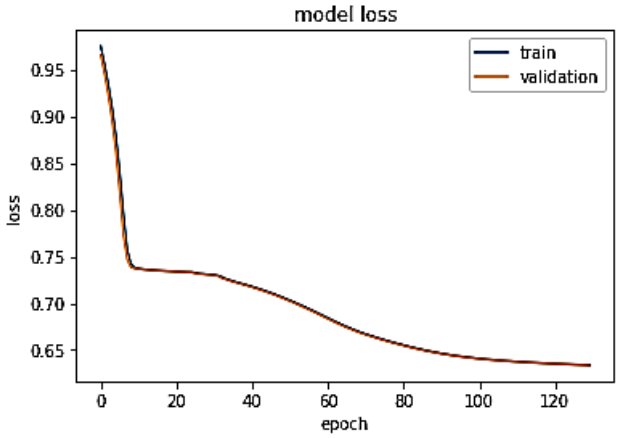


**Fig. 8**: The 5, 000 sample result

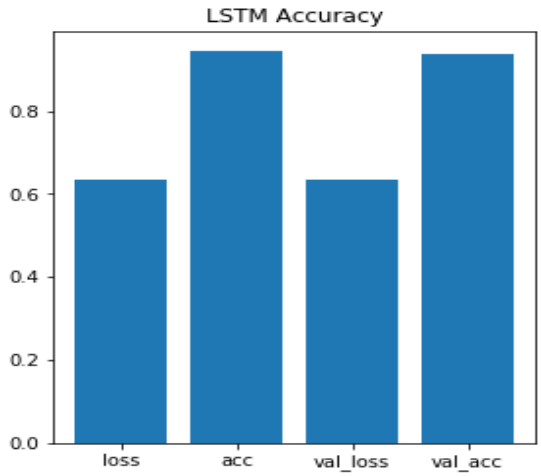


**Fig. 9**: **LSTM model loss and accuracy comparison**

### C. Comparison with Back-Propagation (BP) Algorithm

Back-Propagation (BP) algorithm is used for the classification and regression problems where it is a feed-forward neural network from the expertise of Artificial Neural Networks (ANN) [28]. Feed-forward ANN is inspired by the neuron cells inside the human brain. A neuron will accept input information by dendrites, and then the information will be passed down to synapses. The main idea of this algorithm is to model a given function by changing the internal weightings of the input information to produce expected output information [28]. In this experiment, using the Drebin dataset, the researchers modify it by selecting only 35 samples with 23 features. This is due to the limited time constraint. The hyper-parameter for the BP algorithm is shown in the Table 3 below. The hyper-parameter for the LSTM is still the same as the previous experiment.

**Table 3**: Hyper-parameter for the BP algorithm

| Hyper-parameter | Number |
|---|---|
| Neurons inside hidden layer | 5 |
| Neurons inside output layer | 3 |
| Epochs | 500 |
| Learning rate | 0.3 |

The BP algorithm achieves the average classification accuracy on each fold together with the average performance for all the folds as its output. The BP algorithm final output for the final accuracy accomplished is 82.857%. As a result, the BP algorithm with the chosen configuration had achieved a classification accuracy of 82.857% which is below than the LSTM accuracy of 93.60%. The results from the LSTM accuracy experiment with 93.60% was used from the previous experiment as comparison with the BP algorithm. Figure 10 shows the accuracy comparison between the LSTM and BP algorithms.

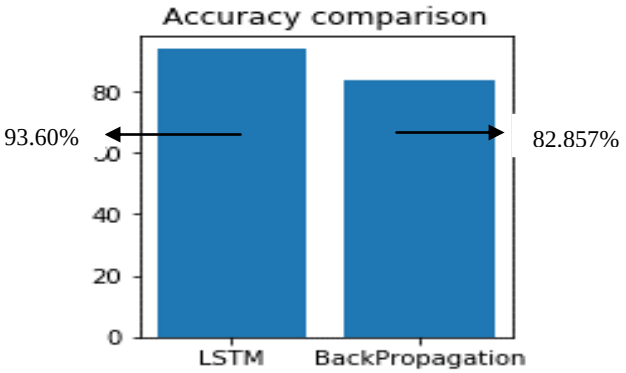


**Fig. 10**: Accuracy comparison of the LSTM and BP algorithms

Therefore, it can be seen that the LSTM performed well rather than the other ANN algorithm. These results from the three experiments reflects that although they have higher rate of accuracy, it still needed to be trained numerous time and also required a huge amount of time for the training and validation processes.

### D. Discussions

Utilizing the details provided by "DroidDetector: Android Malware Characterization and Detection Using Deep Learning" research [17], there are three critical issues in the deep-learning-based (LSTM) Android malware detection that being discovered:

a) *Deep Learning Model*:

In this study, the Long-Short Term Memory (LSTMs) technique is selected because it can train itself more precisely and can produce a high accuracy of malware detection.

b) *Features:*

In the Drebin dataset, there are approximately 54 features that can lead to better categorization and detection of the Android malware. The LSTM algorithm utilizes the features to achieve high accuracy for detecting the malware.

c) *Training Samples:*

The more types of training samples learned, the better the accuracy that can be produced by the algorithm [17]. However, the process of collecting malware dataset was complicated as there are dataset that were protected by password even though it can be directly downloaded from the Internet. This study is extremely crucial for the cyber-security field. The world is getting more and more modern as the time goes by. People worldwide uses their digital world to conduct their daily chores such as bank transactions, bill payment, online shopping, and many more. For the people in the business area, they invest millions of dollars in the stock market. To ensure that their credential information did not fall in the wrong hand, a powerful and effective anti-virus is needed to block all the threat that can jeopardize their efforts in the business world. Based on the accuracy result of 93.60%, it can be proven that the utilization of deep learning in the cyber-security area that specifically targeting on malware detection is crucial. Many researches that have been done on this topic had resulted in excellent result where all of them are above 90% of accuracy rate detection. For example, an accuracy of 94% is achieved in [29], 97.5% in [30], and 97.3% in [31].

## IV. CONCLUSION

In this paper, A Deep Learning Approach to Malware Detection in Android Platform was proposed to achieve a high accuracy metric result by using the Long-Short Term Memory (LSTM) algorithm. Few sets of different experiments were conducted to evaluate the LSTM algorithm capability where as a result, the utilization of 5,000 samples and an accuracy of 93.60% were effectively achieved as the output of these experiments. Additionally, comparative study also has been conducted between the LSTM and the Back-Propagation (BP) algorithm to show the algorithm's capability. The results show that the LSTM algorithm is able to train and validate the Drebin dataset with a better accuracy than the BP algorithm. The LSTM algorithm motivates researchers to utilize the algorithm to enhance the malware accuracy detection. The emergence of Deep Learning (DL) in the Android gadgets anti-malware security has reached factual triumph. Up until now, DL already has been used broadly in the computer science courses for sound, figure and face identification. It is able of automatically discover equivalence in the information, thus making it a rosy approach for the next era of invasion identification. Thus, that is the reason why DL is of eloquent attraction in malware identification for the Android platform on account of it is anticipated to be agile to acquire information of detecting growing malevolent apps that the classic malware identification methods cannot deal with. Furthermore, DL also can be used to precisely discover zero-day assaults and hence the researchers can get a soaring identification tempo [32]. More experiments will be carried to do more comparisons with other standard algorithms to evaluate the performance of LSTM algorithm.

For the future works, the LSTM algorithm should have the capability to produce a high accuracy for detecting malicious file types such as URLs, PDFs and other possible medium for malware to resides as well as software applications that can run in the Android platforms. Apart of that, further enhancement of the present LSTM algorithm is needed to produce a higher accuracy.

## ACKNOWLEDGMENT

This research is supported and funded by RIMC of University Malaysia Sarawak (UNIMAS), under the special Grant Scheme (F04/SpGS/1547/2017).

## REFERENCES

1. Black M (2018, Apr 16), How to Remove Android Virus in 3 Simple Steps. TECH ADVISOR FROM IDG. Retrieved from https://www.techadvisor.co.uk/how-to/google-android/remove-android-virus-3633110/
2. Lookout (2018), Should You Worry About Getting a Cell Phone Virus? Lookout. Retrieved from https://www.lookout.com/know-your-mobile/android-virus
3. Radicati S (Ed.), Mobile Statistics Report, 2016-2020 (2016). Retrieved from http://www.radicati.com/wp/wp-content/uploads/2016/01/Mobile-Growth-Forecast-2016-2020-Executive-Summary.pdf
4. Egele M, Scholte T, Kirda E & Kruegel C (2008), A Survey on Automated Dynamic Malware Analysis Techniques and Tools. *ACM CSUR* 44(2): 6:1-6:42.
5. Xu J, Yu Y, Chen Z, Cao B, Dong W, Guo Y, Cao J (2013), MobSafe: cloud computing based forensic analysis for massive mobile applications using data mining. In *Tsinghua Sci. Technol*. 18: 418–427.
6. Felt AP, Finifter M, Chin E, Hanna S & Wagner D (2011), A survey of mobile malware in the wild. In Proceedings of the *1st ACM workshop on SPSM*, pp. 3-14.
7. Zhou Y & Jiang X (2012), Dissecting android malware: Characterization and evolution. In Proceedings of *2012 IEEE Symposium on Security and Privacy (IEEE S&P 2012)*, pp. 95–109.
8. G Data: Mobile malware report: Q2/2015. Retrieved from https://public.gdatasoftware.com/Presse/Publikationen/Malware_Reports/G_DATA_M obileMWR_Q2_2015_EN.pdf. Accessed 15 July 2016
9. Symantec: Internet security threat report, volume 20. Retrieved from https://www.symantec.com/security_response/publications/threatreport.jsp Accessed 15 July 2016
10. Tam K, Feizollah A, Anuar NB, Salleh R & Cavallaro L (2017), The evolution of android malware and android analysis techniques. *ACM Comput. Surv*. 49: 76:1– 76:41.
11. Kaspersky Lab (2015), The Great Bank Robbery. Retrieved from http://www.kaspersky.com/about/news/virus/2015/Carbanak-cybergang-steals-1-bn-USDfrom-100-financial-institutions-worldwide, 2015
12. CIS (2017), Top 10 Malware of April 2017. *Center for Internet Security*. Retrieved from https://www.cisecurity.org/top-10-malware-of-april-2017/
13. DI (2017, Jan 19), 7 Biggest Malware Threats of 2017: Android Malware. *Defence Intelligence*. Retrieved from https://defintel.com/blog/index.php/2017/01/7-biggest-malware-threats-of-2017.html
14. Hor RD (2017), Deep Learning: With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart. *MIT Technology Review*. Retrieved from https://www.technologyreview.com/513696/deep-learning/
15. Santos LA (n.d.), Artificial Intelligence: Introduction. *Gitbooks*. Retrieved from https://leonardoaraujosantos.gitbooks.io/artificialintelligence/content/deep_learning.html
16. McLaughlin N, Martinez del Rincon J, Kang B, Yerima S, Miller P, Sezer S, Safaei Y, Trickel E, Zhao Z, Doupe A & Ahn GJ (2017), Deep Android Malware Detection. Retrieved from https://dl.acm.org/citation.cfm?id=3029823
17. Yuan Z, Lu Y & Xue Y (2016), DroidDetector: Android Malware Characterization and Detection Using Deep Learning. *TSINGHUA SCIENCE AND TECHNOLOGY* 21(1): 114-123. ISSN 1007-0214.

18. Yuan Z, Lu Y, Wang Z & Xue Y (2014), Droid-Sec: Deep Learning in Android Malware Detection. Proceedings of the *2014 ACM Conference on Special Interest Group on Data Communication (SIGCOMM, poster)*, pp. 371–372.
19. Nauman M, Tanveer TA, Khan S & Syed TA (2017), Deep neural architectures for large scale android malware analysis. *Cluster Computing*: 1-20. Springer Science+Business Media New York 2017. DOI 10.1007/s10586-017-0944-y
20. Olah C (2015, Aug 27), Understanding LSTM Networks. Colah's blog. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/
21. Sundermeyer M, Schluter R & Ney H (2012), Lstm neural networks for language modelling. In *INTERSPEECH*, pp. 194-197
22. Gers FA & Schmidhuber J (2001), LSTM recurrent networks learn simple context free and context sensitive languages. *IEEE Transactions on Neural Networks* 12(6): 1333-1340.
23. Athiwaratkun B & Stokes JW (2017, June 19), Malware classification with LSTM and GRU language models and a character-level CNN. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2482-2486. Doi: 10.1109/ICASSP.2017.7952603
24. Xiao X, Zhang S, Mercaldo F, Hu G & Sangaiah AK (2017, Sept 2), Android malware detection based on call sequences and LSTM. SpringerLink. Retrieved from https://link.springer.com/article/10.1007/s11042-017-5104-0
25. Gibson A & Patterson J (2017), Chapter 4. Major Architectures of Deep Networks. *Safari*. Retrieved from https://www.safaribookonline.com/library/view/deeplearning/9781491924570/ch04.html
26. Prk54. (2018). Drebin Dataset. Retrieved from https://github.com/prk54/malware-detection-machine-learning-approach
27. Institute for System Security. (2016). The Drebin Dataset. Retrieved from https://www.sec.cs.tu-bs.de/~danarp/drebin/
28. Brownlee J (2016, Nov 7), How to Implement the Backpropagation Algorithm from Scratch in Python. *Machine Learning*. Retrieved from https://machinelearningmastery.com/implement-backpropagation-algorithm-scratchpython/
29. Rhode M, Burnap P & Jones K (Dec 2, 2017), Early-Stage Malware Prediction Using Recurrent Neural Networks. School of Computer Science and Informatics, Cardiff University. Retrieved from https://arxiv.org/pdf/1708.03513.pdf
30. Su X, Zhang D, Li W & Zhao K (2016), A Deep Learning Approach to Android Malware Feature Learning and Detection. In *2016 IEEE TrustCom/BigDataSE/ISPA*, pp. 244-251. Doi: 10.1109/TrustCom.2016.0070
31. Fereidooni H, Conti M, Yao D & Sperduti A (2016), ANASTASIA: Android Malware Detection Using Static Analysis of Applications. *New Technologies, Mobility and Security (NTMS), 2016 8th IFIP International Conference*, pp. 1-5. Doi: 10.1109/NTMS.2016.7792435
32. Tang TA, Mhamdi L, McLernon D et al. (2 more authors) (2016), Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. In: *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM).* The International Conference on Wireless Networks and Mobile Communications (WINCOM'16), 26-29 Oct 2016, Fez, Morocco. IEEE. ISBN 978-1-5090-3837-4. Retrieved from https://doi.org/10.1109/WINCOM.2016.7777224