Detection Of Malware Using Deep Learning Techniques

Garminla Sampath Kumar, Pooja Bagane

Abstract: Malware continues to be a serious threat starting from home users to large enterprises. This makes it a hot research topic. Detection of malware is done using static and dynamic analysis of malware signatures and behavior patterns. These are proven to be ineffective and time consuming while detecting unknown malware. In order to identify the new malware many machine learning algorithms are created. Feature engineering is a key step for building these algorithms. This takes too much time. By using deep learning techniques this step can be completely avoided. Recent research reported that many of them used biased dataset, which is completely ineffective in real-time situations. Hence this drives to create a new algorithm/architecture to detect malware using deep learning. By using specialized Convolutional Neural Networks for capturing patterns in malware sequences using the concept of weight sharing. Also adding this with Recurrent Neural Networks, we could capture recurring patterns in malware.

Index Terms: Malware analysis, Convolutional Neural Networks, Recurrent Neural Networks, image processing.

1 INTRODUCTION

Malware is a major threat to the security of computer users which can cause huge financial losses to firm. With increasing applications of Internet of Things (IoT), this made attackers to target them. Malware has different names such as adware, rootkit, backdoor, ransomware, trojans, worms, spyware etc. i.e depending on the behavior, thus detecting these malwares became as an evolving problem for researchers. There are two types of malware analysis and detection mechanisms: static analysis and dynamic analysis. Examining and Extracting information from theexecutable file without running is Static Analysis. Running the malware and observing its behavior on the system is Dynamic analysis. With a new variant of malware, experts generally analyze the sample manually or create a program that can match with similarity of this class of malware. Recently, image classification has been improved a lot with the development of deep learning techniques. Convolutional Neural Networks demonstrated better performance. Here feature engineering, feature learning and feature representation are automatically acquired. This paper includes the related works and algorithms in section 2, design methodology in section 3 and conclusion in section 4.

2. RELATED WORK

Malware can be detected by finding a match with virus definition dataset which is updated from time to time. This method is signature-based detection. However, when a new malware variant is found, it couldn't detect it [1]. However this requires extensive domain knowledge to reverse engineer. In order to avoid this detection, hackers use polymorphism as obfuscation techniques. Many software tools can be used to unpack this. As this process is resource intensive task, [2] presented this as a 3step process.

- *Garminla Sampath Kumar is currently pursuing masters degree program in Computer science engineering in Symbiosis International University, India, E-mail: sampathreddy.garminla.mtech2018@sitpune.edu.in*
- *Pooja Bagane is currently working as assistant professor in the department of computer science engineering in Symbiosis International University, India, E-mail: pooja.bagane@sitpune.edu.in*

In the first step, malware is unpacked. In step 2, the executable is disassembled. In step 3, API call is extracted. Step 4 involves API call mapping and statistical feature analysis. [3] presented an extension to [2] by adding another step. Here it involves incorporating machine learning techniques. Recently with the increase in malware attacks and obfuscated malware [4], many researches are improving machine learning algorithms for malware detection. Machine learning Algorithms (MLAs) requires feature engineering and feature selection. It requires domain level knowledge. Various features can be obtained through Static and Dynamic analysis. Static Analysis can be done by capturing the information from executables without running it. Dynamic analysis is done by running the malware in an isolated environment. Various complexities of Dynamic Analysis explained in [5]. Dynamic Analysis is efficient and long-term solution for malware detection. But deploying it would take a long time to analysis the executable. Anti-malware commercial programs generally use the hybrid of static and dynamic analysis. In recent times, Deep learning is improved a lot. Here feature engineering is not required because it will learn them automatically. It also outperformed many machine learning algorithms. There exist a very few research studies towards the application of deep learning for malware analysis. As Dynamic Analysis requires a specialized environment, here I would concentrate only on Static Analysis. Most of the new malwares are formed by changing a small part of old malware to evade signature-based detection. To overcome this, it is important to learn the local and similar characteristics of malware. Several malwares have only small change in code, we could take advantage of this by using image processing techniques. Once the malware binary is converted to image and applying classification algorithms on this makes the task simple. As only a small part of code is changed, only a small part of the image is changed. Global view of the image remains the same. The main advantage of the image conversion is that it can handle the packed malware without even unpacking it. Convolutional Neural Network and Recurrent Neural Networks can be used. CNN is used on images. RNN can be used after flattening the last layer of CNN. RNN is used to detect recurring patterns.

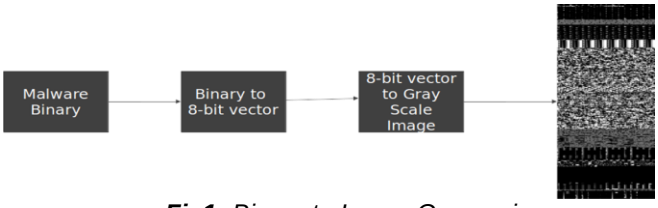

Fig1: *Binary to Image Conversion*

In order to convert to image, we need to decide the width of the images. The following table shows, how to decide it. [6]

File Size Range	Image Width
<10 kB	32
10 kB – 30 kB	64
30 kB – 60 kB	128
60 kB – 100 kB	256
100 kB – 200 kB	384
200 kB – 500 kB	512
500 kB – 1000 kB	768
>1000 kB	1024

Table1: *Width of Images*

2.2 Convolutional Neural Network

CNN is a deep learning model which has shown successful results particularly in the field of image classification. CNN consists of multilayered neural networks as well as other deep learning methods and has a specific structure called a convolution layer.

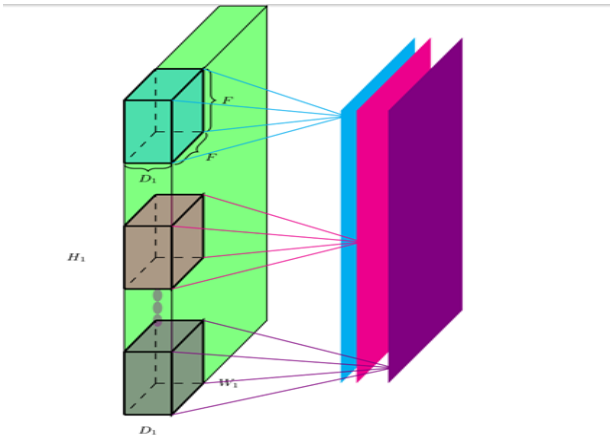

Fig2: *Width, Height and Depth of CNN*
Source: http://cs231n.stanford.edu/

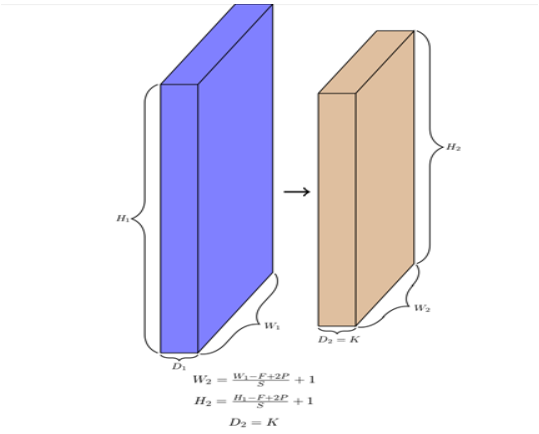

Fig3: *Depth of the output*
Source: http://cs231n.stanford.edu/

From Fig2, we define the following quantities Width (W1), Height (H1) and Depth (D1) of the original input. The Stride S, the number of filters K, the spatial extent (F of each filter (the depth of each filter is same as the depth of each input). The output is W2 × H2 × D2. From Fig3, K filters will give us K such 2D outputs. We can think of the resulting output as K × W2 × H2 volume. Thus D2 = K. Fig4 is a sample CNN for handwritten character recognition.

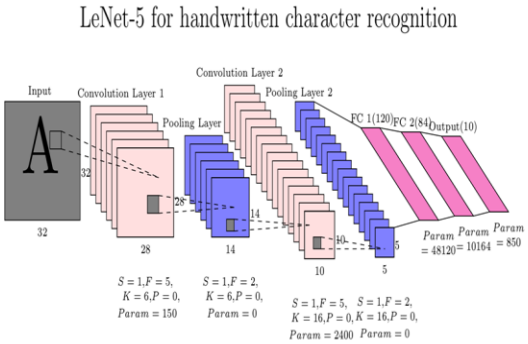

Fig4: *CNN for handwritten character recognition*
Source: http://cs231n.stanford.edu/

2.3 Recurrent Neural Network:

RNNs are neural networks, specialized for processing a sequence of values x (1) , . . . , x (τ) . It can scale to much longer sequences that would be practical for networks without sequence-based specialization. It does so by sharing parameters across different parts of a model. Fig5 explains recurrent connections between hidden units. and Fig6 explains RNN withsingle input and single output.

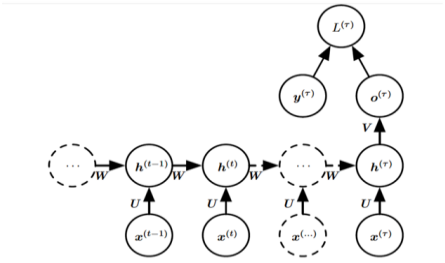

Fig5: *Recurrent connections between hidden units*
Source: http://cs231n.stanford.edu/

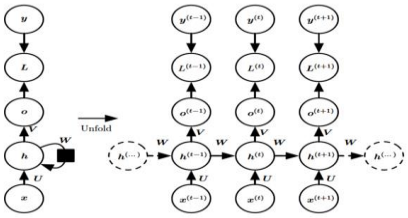

Fig6: *RNN with single input and single output*
Source: http://cs231n.stanford.edu/

2.4 Malimg dataset

Malimg dataset contains 9,339 malware samples from 25 various malware families. The detailed statistics of the dataset is reported following table.

No.	Class	Family Name	No. of Variants
1	Worm	Allaple.L	1591
2	Worm	Allaple.A	2949
3	Worm	Yuner.A	800
4	PWS	Lolyda.AA 1	213
5	PWS	Lolyda.AA 2	184
6	PWS	Lolyda.AA 3	123
7	Trojan	C2Lop.P	146
8	Trojan	C2Lop.gen!G	200
9	Dialer	Instantaccess	431
10	Trojan Downloader	Swizzor.gen!I	132
11	Trojan Downloader	Swizzor.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Alueron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	Trojan Downloader	Wintrim.BX	97
19	Dialer	Dialplatform.B	177
20	Trojan Downloader	Dontovo.A	162
21	Trojan Downloader	Obfuscator.AD	142
22	Backdoor	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Skintrim.N	80

Table2: *Statistics of the data sets*

3. DESIGN METHODOLOGY

First malware binary is converted into image. Apply CNN on the image. As last layer in CNN is in form of n-dimentional matrix, we will flatten it. Flattening means we convert n-dimentional matrix to 1-dim column matrix. Flattening is done because the Bi directional Long-Short Term Memory (BiLSTM) requires single dimentional input. Flattened input is passed into BiLSTM. BiLSTM will try to learn the features by trying to classify the input dataset.

1. Converting Malware binary to Image.
2. Applying CNN to Malware Image.
3. Converting CNN into Flatten.
4. Applying BiLSTM to classify the input.

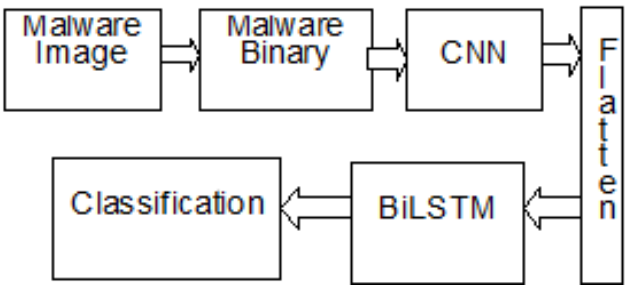

Fig7: *BiLSTM system*

4 CONCLUSION

Most of the people used MLAs. Applying Deep Learning architectures can give us good results compared to MLAs. If the number of layers in these architectures are more, then the amount of time to train these will increase. Similarly test time complexity increases. Hence I would like to design a Light-weight Malware Classifier which can tell whether the file is malware or not.

5 REFERENCES

[1] Li, Bo, et al. "Large-scale identification of malicious singleton files." Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. ACM, 2017.

[2] Alazab, Mamoun, Sitalakshmi Venkataraman, and Paul Watters. "Towards understanding malware behaviour by the extraction of API calls." 2010 Second Cybercrime and Trustworthy Computing Workshop. IEEE, 2010.

[3] Tang, MingJian, Mamoun Alazab, and Yuxiu Luo. "Big data for cybersecurity: Vulnerability disclosure trends and dependencies." IEEE Transactions on Big Data (2017). to be published.

[4] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, and A. Alazab, "Cyber-crime: The case of obfuscated malware," in Global Security, Safety and Sustainability & e-Democracy (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 99, C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-Nemrat, Eds. Berlin, Germany: Springer, 2012.

[5] Rossow, Christian, et al. "Prudent practices for designing malware experiments: Status quo and outlook." 2012 IEEE Symposium on Security and Privacy. IEEE, 2012.

[6] Nataraj, Lakshmanan, et al. "Malware images: visualization and automatic classification." Proceedings of the 8th international symposium on visualization for cyber security. ACM,

2011.
[7] Su, Jiawei, et al. "Lightweight classification of IoT malware based on image recognition." 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2018.
[8] Vinayakumar, R., et al. "Robust Intelligent Malware Detection Using Deep Learning." IEEE Access 7 (2019): 46717-46738.
[9] Agrawal, Rakshit, et al. "Attention in Recurrent Neural Networks for Ransomware Detection." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.
[10] Naeem, Hamad, Bing Guo, and Muhammad Rashid Naeem. "A light-weight malware static visual analysis for IoT infrastructure." 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, 2018.
[11] Li, Bo, et al. "Large-scale identification of malicious singleton files." Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. ACM, 2017.
[12] Nataraj, Lakshmanan, et al. "Sarvam: Search and retrieval of malware." Proceedings of the Annual Computer Security Conference (ACSAC) Worshop on Next Generation Malware Attacks and Defense (NGMAD). 2013.
[13] Tobiyama, Shun, et al. "Malware detection with deep neural network using process behavior." 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2016.
[14] Anderson, Hyrum S., et al. "Evading machine learning malware detection." Black Hat (2017).
[15] Pascanu, Razvan, et al. "Malware classification with recurrent networks." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.
[16] Raff, Edward, et al. "An investigation of byte n-gram features for malware classification." Journal of Computer Virology and Hacking Techniques 14.1 (2018): 1-20.
[17] Bailey, Michael, et al. "Automated classification and analysis of internet malware." International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg, 2007.
[18] Alam, Mohammed S., and Son T. Vuong. "Random forest classification for detecting android malware." 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing. IEEE, 2013.