

Classifying Malware Using Convolutional Gated Neural Network

Chang Hoon Kim*, Espoir K. Kabanga, Sin-Jae Kang

Department of Computer and Information Engineering, Daegu University, Gyeongsan, Republic of Korea

kimch@daegu.ac.kr, kabanga@daegu.ac.kr, sjkang@daegu.ac.kr

Abstract— Malware or Malicious Software, are an important threat to information technology society. Deep Neural Network has been recently achieving a great performance for the tasks of malware detection and classification. In this paper, we propose a convolutional gated recurrent neural network model that is capable of classifying malware to their respective families. The model is applied to a set of malware divided into 9 different families and that have been proposed during the Microsoft Malware Classification Challenge in 2015. The model shows an accuracy of 92.6% on the available dataset.

Keywords— Malware, Machine Learning, Neural Network, Deep Neural Network, CNN, Gated Recurrent Unit

I. INTRODUCTION

More than 4 billion people around the globe own a computer or an electronic device capable of providing them access on internet and more than 70% of human activities have been now computerized. These numbers do not stay static but keep increasing proportionally to time. The increase in number of users of computers and other electronic devices and the huge increase of access to internet connection (from less than 1% in 1995 to around 40% today according to internetlivestats.com) have led to different types of cybercrimes.

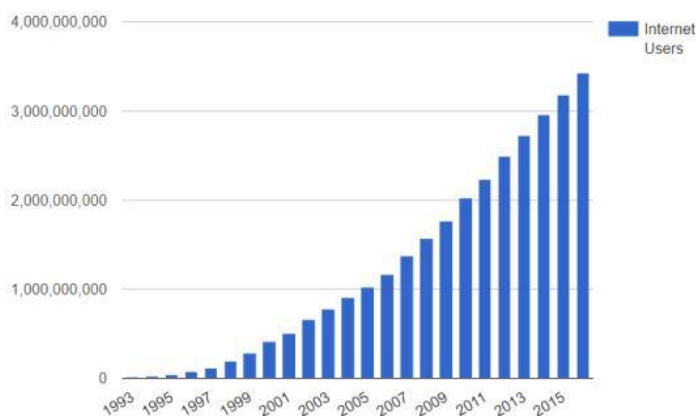


Figure 1. Number of internet Users [9]

Cybercrime is mostly done through malware attack, which

infect a computer or an electronic device in order to perform the malicious task that its creator has assigned to it. Based on the different types of malicious task of malware into an infected system, we can classify malware into different types:

- **Virus:** this is a program that is designed to infect a software program. Once it is installed on a program that is run by a user, it is capable of spreading itself to other computers and cause severe malfunction to other files in the system.
- **Worm:** It is a program that is capable of spreading itself through a network directly. It is a stand-alone program, therefore it does not need to attach itself to any existing program. The main difference between a virus and a worm is that a virus needs to be activated by a user for it to be spread while a worm does not need any human effort to replicate or spread itself.
- **Trojan:** This type of malware is commonly called trojan horse. It got its name to the Greek story. It is a type of malware that is thought to be or appear to be a normal file or program and persuade to the user to use it into the system. Once it is installed in the system, it enables unauthorized access to the system and therefore there can be theft or loss of information, unauthorized modification of data and several other dangerous activities without knowledge of the system owner.
- **Adware:** An adware is a type of malware that directs a computer user to unwanted types of advertisements. Usually, these advertisements lead to the download of malicious files or to the websites with sexual contents.
- **Spyware:** Once in the system, this type of malware is able to monitor the user's activities without their knowledge and consent. Among the spying activities, there is collecting keystrokes, data harvesting

* Corresponding Author

(financial data, usernames and passwords, etc.) and many more. Sometimes, a spyware can bundle itself with a trojan horse or any other software in order to spread itself.

- **Rootkit:** This is a type of malware can remotely access or control a system without being detected by its owner or a security program. A rootkit is capable of modifying a system configuration, stealing or accessing information, installing other types of malware or controlling the system as part of the botnet. It is difficult to prevent or to detect the rootkit due to their hidden nature.
- **Ransomware:** This type of malicious software kind of takes the computer system hostage until a ransom is paid to the attacker. It restricts access to the computer by encrypting files on the hard drive or locking down the system while it displays a message demanding the user to pay a certain amount of money, usually through bitcoins, for the system to get released.
- **Backdoor:** a backdoor is designed to secure unauthorized remote access to a computer or to gain access to plaintext in cryptographic system. It is created to compromise a system and its normal authentication procedures.
- **Botnet:** it is created to specifically perform some automatic operations. It is similar to the spyware but the different is that the botnet does not collect information as does the spyware. Instead of that, it waits until it receives a command from the attacker through a command-and-control server. A botnet is usually designed for DDoS attacks and it exploit the security vulnerabilities to infect in order to infect other computers and devices.

The malware industry keeps growing year by year. Every year, thousands of new malware are created and this constitutes a big challenge for anti-malware building industries to be able to develop effective solutions to counter the malicious effects of the malware files into the their customers systems.

Being able to distinguish and to classify different types of malware is very important and can help to have a clear view on the way they can infect systems.

II. RELATED WORK

Machine learning techniques have taken a very important place for the task of malware classification. These techniques have proven to be more efficient and effective compared to traditional classification techniques.

In [1] they proposed a recurrent neural network approach to classify malware file, where they learned the language of malware that is spoken through the executed instructions. That approach, which is similar to natural language model, provided them a good rate of true positive and false positive. [5] proposed new malware classification architecture that included LSTM language model and GRU model language model. They also proposed in their work a new single-stage malware classifier based on character-level convolutional neural network and they obtained 31.3% improvement in true positive rate in comparison to the work performed in [1] at a false positive rate of 1%. Other works like the one done in [2] and [3] uses convolutional neural network to classify malware and their work used a technique developed in the research done by Nataraj et al. in [4] consisting of visualizing malware data. [2] won the 2015 Microsoft malware classification challenge thanks to their solution with relied on the extraction of opcode 2,3 and 4 grams features, segment line count features and asm file p [7] used Deep Neural Network approach to classify malware.

III. PROPOSED MODEL

The model proposed to classify malware is inspired from the model used in [10] for audio tagging. Our challenge is to apply this architecture to malware dataset and check the obtained accuracy. The overall architecture is described in Figure 2.

The model has four distinct levels: a convolutional neural network (CNN), a layer of gated recurrent units (GRUs), a layer of deep neural networks (DNNs), and a sigmoid layer.

Each convolutional neural network has three sublayers: a convolution layer, an activation layer, and a pooling layer. In the convolution layer, a user-defined or randomly-generated filter is convolved with the input matrix to produce a second matrix. The values of this second matrix are passed to the activation layer, where ReLu, tanh, or sigmoid activation functions are applied. The results of this layer are passed to a pooling layer where max pooling or average pooling is applied to produce the output value of the CNN.

The GRU layer consists of a user-defined number of GRU units. Each CNN in the previous layer produces a single

output value, and each GRU in the GRU layer receives the values of all CNNs and treats them as a time series from lowest number to highest number. Each GRU produces a single output value per input value, resulting in a number of output values equal to the number of CNNs in the previous layer. The randomly-generated weights internal to the GRUs (unique to each GRU) ensure that all GRUs will not produce the same value for a given input dataset.

The GRUs are gating mechanism which were introduced in 2014 by Junyoung et al. [8] and which are very similar to the Long Short-Term Memory (LSTM) expect the fact that they have fewer parameters than LSTM and the lack an output gate.

The third layer of the neural network is a network of DNNs. There are an equal number of DNNs as GRUs and each receives only the outputs of its corresponding GRU. These inputs are weighted, summed and passed through an activation node using the ReLu, tanh, or sigmoid activation functions. Each DNN node produces a single output value, so the entire DNN layer produces a vector of values equal in length to the number of GRU or DNN nodes.

The final layer of the neural network is a sigmoid layer consisting of a single node that receives the outputs of all DNN nodes. The vector of inputs is weighted and summed similar to the previous layer then passed through a sigmoid activation function. The result of this is the output classification of the neural network.

In the training phase, the user can provide a “truth” value to the neural network. The difference between this and the output classification is an error that can be used in backpropagation to train the weights of the GRU, DNN, and sigmoid layers. Due to the nature of the pooling function of the CNN layer, it’s not possible to train the filters at that level, so testing with different values or generating “optimal” ones based off the Convolutional Neural Networks paper is recommended.

For the CNN layer, the number of CNNs, filter size and values, activation function, and pooling type are user-definable values. The number of CNNs determines the number of slices into which the input dataset will be divided. The width of the input matrix to each CNN is calculated as the ceiling of the total width of the input matrix divided by the number of CNNs. If the original input matrix does not divide evenly, the last and second to last CNNs will share some columns of the original input matrix.

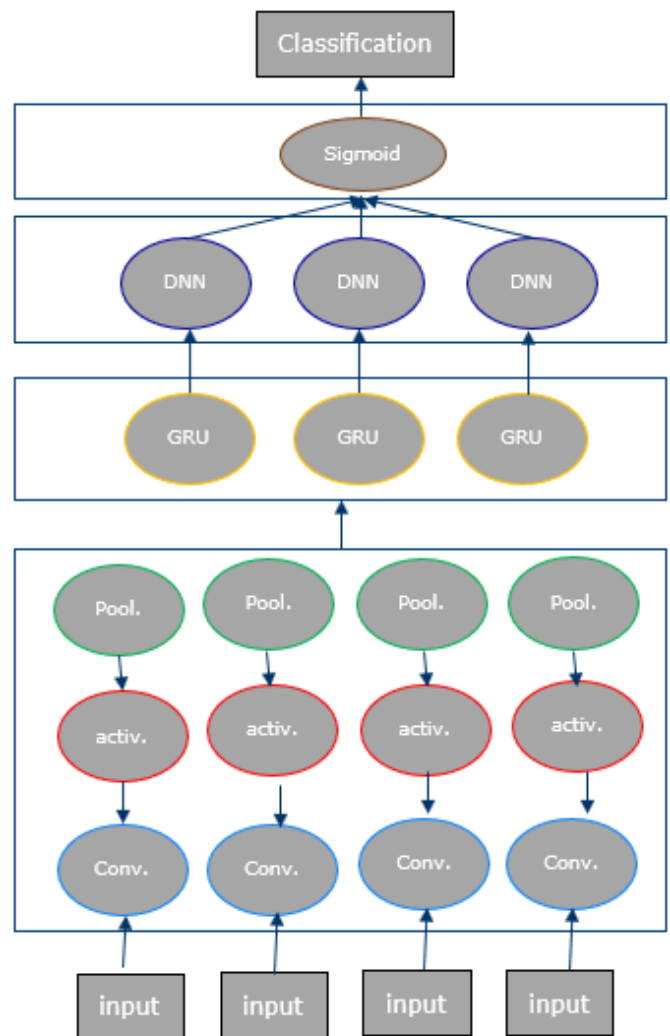


Figure 2. Proposed Architecture

IV. DATASET

The dataset is consisted of data collected from Kaggle [13], which is a platform where a large community of data scientists compete to solve valuable problems. Microsoft organized a competition in 2015 through Kaggle platform with aim of malware classification.

The dataset consists of 9 different families of malwares and 21741 samples among which 10868 are for training and 10873 for testing. For each file, the raw data contents the hexadecimal representation of file’s binary content (byte files), without PE header. Microsoft provided also a metadata manifest, which is a log containing various metadata information extracted from the binary, such as function calls, string, etc., and this was generated using the IDA disassembler tool (ASM files).

For this work, we only use the binary data.

The malware 9 families to which the malware data belong are: 1 - Ramnit, 2 - Lollipop, 3 - Kelihos_ver3, 4 - Vundo, 5 - Simda, 6 - Tracur, 7 - Kelihos_ver1, 8 - Obfuscator.ACY, 9 - Gatak.

Figure 3 describes the used dataset.

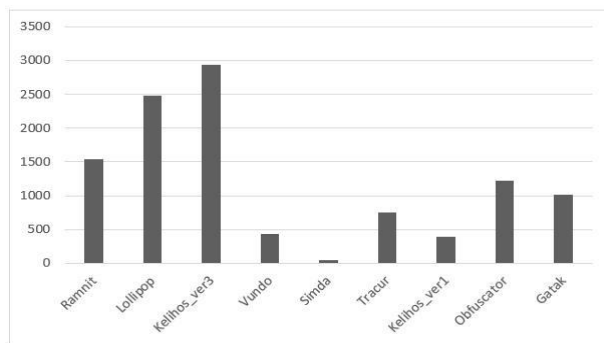


Figure 3. Dataset

V. EXPERIMENTAL RESULTS

The neural network has two possible running states: building and classifying. In the build state, all the settings discussed in section III are set by the user and a script creates a neural network, which is saved to a file. In the classify state, the user provides a neural network file, an input file, and optionally a truth value. The network provides a classification for the input and, if a truth value is provided, updates its weight vectors based on the classification and saves the updated network to the network file.

Instructions are extracted from malware binaries and are used as parameters. For each of the instructions to be extracted, a "one-hot" vector is created where there is a value for each possible x86 instruction (and an unknown instruction), where all values except the one for the mnemonic of the target instruction are zero and the value for the target instruction is a one. By running this script against a dataset of malware, it is possible to generate a dataset of input files for the malware for use with the neural network. The number of instructions extracted per file should (but is not required to) be the same for all inputs to the same neural network, but could be varied with different neural networks for testing different variables.

This model achieves 92.66% accuracy.

VI. CONCLUSION AND FUTURE WORK

In this project, we applied convolutional gated recurrent neural network to malware classification. The model combined, that was built based on a four layers architecture, includes a CNN layer, a GRU layer, a DNN layer and a sigmoid layer and this achieved 92.66% classification

accuracy on binary data provided by Microsoft during the malware classification challenge of 2015.

Achieved result for this work is considered as state-of-the-art but effort needs to be made in order to improve this accuracy.

ACKNOWLEDGMENT

This research was supported by the Technology Transfer and Commercialization Program through INNOPOLIS Foundation funded by the Ministry of Science and ICT, Republic of Korea. (2017DG0018 / Intelligence CCTV Using 3D Camera)

REFERENCES

- [1] Razvan Pascanu, Jack W. Stokes, Hermineh Sanossian, Mady Marinescu, Anil Thomas "Malware classification with recurrent networks" IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015
- [2] Xiaozhou Wang, Jiwei Liu, Xueer Chen, "Say No to Overfitting", Winner of Microsoft Malware Classification Challenge (BIG 2015)
- [3] Daniel Gibert "Covolutional Neural Networks for Malware Classification", Master in Artificial Intelligence thesis, University of Barcelona, 20th October 2016
- [4] Lakshmanan Nataraj, S. Karthikeyan, G. Jacob, B. s. Manjunath, "Malware Images: Visualization and Automatic Classification", Proceedings of the 8th International Symposium on Visualization for Cyber Security, Article No. 4, 2011
- [5] Ben Athiwaratkun, Jack W. Stokes, "Malware Classification with LSTM and GRU Language Models and Character-Level CNN", 2017 IEEE International Conference on Acoustics, Speech and Signal Processing
- [6] Malware Classification Challenge
"https://www.kaggle.com/c/malware-classification"
- [7] Joshua Saxe and Konstantin Berlin, "Deep neural network based malware detection using two dimensional binary program features," arXiv preprint arXiv:1508.03096v2, 2015.
- [8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks no Sequence Modeling", arXiv:1412.3555v1, 11 Dec 2014 [cs.NE]
- [9] <http://www.internetlivestats.com/internet-users/>
- [10] Yong Xu, Q. Kong, Q. Huang, W. Wang, M.D. Plumley, "Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging", arXiv:1702.07787v1 [cs.SD] 24 Feb 2017



Mr. Chang Hoon Kim is an associate professor at Daegu University, Republic of Korea. He earned his bachelor of computer science engineering in 2001, a master degree of computer and information engineering in 2003 and a Ph. D in computer science engineering in 2006, both at Daegu University in the Republic of Korea. Professor Kim has published many papers in international and domestic journals and has attended many international and domestic conferences for presentations. His research area includes network security, system security and artificial intelligence.



Mr. Espoir K. Kamundala is master degree student in computer and information engineering at Daegu University at the Republic of Korea. He earned a bachelor degree in computer applications in 2015 at CMR Institute of Management Studies (affiliated to Bangalore University) in Bangalore, India. His research area includes machine learning, malware detection and classification and forensic science.



analysis.

Dr. Kang is a professor in the department of computer and information engineering at Daegu University, Republic of Korea. He earned his master degree in computer engineering in 1997 and a PhD in computer engineering at Pohang University of Science and Technology in 2002, majoring in Natural Language Processing. His main research area includes natural language processing, machine learning and big data