TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)
7-10 Dec 2021. Auckland, New Zealand

Portable Executable Malware Classifier Using Long Short Term Memory and Sophos-ReversingLabs 20 Million Dataset

Julianne Alyson Diaz, Argel Bandala
Electronics and Communications Engineering Department
De La Salle University
Manila, Philippines
julianne_alyson_diaz@dlsu.edu.ph

***Abstract* - This research paper proposes the Utilization of Long Short Term Memory(LSTM) paired with LightGBM in Portable Executable (PE) Malware Classification, which will be trained and tested with the Sophos-ReversingLabs 20 Million Dataset (SoReL-20M). PE files are regular executable, object codes, and Dynamic Link Libraries (DLLs) files used commonly in Windows operating systems in 32-bit and 64-bit versions. Problems, when PE malware is not detected, is its ability to install rootkits, worms, trojans and etc. Current development in PE malware detection suggests signature-based detection. Although most studies produce high accuracy, it is not always applicable to all scenarios, especially on zero-day attacks. Other studies in malware detection suggest the use of a non-signature-based approach, hence the proposed method of utilization of LSTM for the research. Due to the large number of SoReL-20M dataset to be processed, LightGBM will be used to reduce its impact on the resources.**

Keywords- Portable Executable Malware; Long Short Term Memory; Sophos Dataset;

I. INTRODUCTION

In a person's day-to-day life, the utilization of technology increases with its development and popularity. Every action that we do in these devices can provide data to both help device manufacturers and data science people to improve our usage, at the same time we also may become a target of malicious intentions. Due to our data being available everywhere and can be accessed remotely, the popularity of malicious activity also increases. With a few data extracted from a person, a hacker with malicious intention can instantly launch an attack on either the targeted person, organization, or institution. Data infiltrations are usually undetected until the attacker launches the attack. By then it is already too late, and damages are already done, such as data exploitation, data corruption, or other malicious intentions. All those breaches and damages can lead to identity theft, financial resources lost, security and reputations can be compromised.

Malware is a code that is written for malicious intentions, this malicious intention includes may include reconnaissance attacks, ransomware and etc. The damage, behavior, and goal of malware differ depending on the attacker. Currently, there are two types of malware detection, the signature base, and the non-signature base. In signature-based detection, the signatures of known malware were analyzed to determine whether the file of the machine is infected with malware. In recent studies on signature-based detection, n AI-based malware detection and classifier was used. In the research, experiments were conducted on the 5-Layer stacked autoencoders for he malware feature extraction and signature base training detection. Testing of the model was done on synthetic databases and yielded an accuracy of 95.6%, which is higher than the traditional signature-based method that has an accuracy of 84.6%. [1]

Non-signature-based detection on the other hand was used to detect a zero-day attack, where no known signatures were found. In some cases, zero-day attacks were left undetected. [2] For the non-signature-based detection, Deep Random Forest Paradigm was proposed in detecting zero-day attacks. The study utilized Malimg, BIG 2015, and MaleVis malware datasets, which yielded accuracies of 98.65%, 97.2%, and 97.43%. [3] In another study on non-signature bases malware detection on Portable Executable (PE) files, static analysis was used for feature extraction. For the testing and training, different machine algorithms were used to determine the best method. From the results of the study, kNN yielded the highest accuracy.

TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON) | 978-1-6654-9532-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/TENCON54134.2021.9707353

II. GENERAL SYSTEM ARCHITECTURE

In system design for PE malware classifiers, the following techniques were considered: LSTM and LightGBM. The general workflow diagram of the program can be seen below in figure 1.

On the diagram, the Sophos-ReversingLabs 20 Million Dataset(SoReL-20M) dataset will be loaded. From there hash file sha256 of the dataset was calculated. The malware binaries will then be extracted. Once those features were extracted, dataset fitting and filtering of missing data were done. The dataset with its features extracted will then be split into training and testing datasets. The LSTM will then do the dynamic analysis while a Light GBM will do the static analysis to act as a boost for the LSTM.

III. THEORETICAL DESIGN AND CONSIDERATION

In this section, the main design considerations in this research will be discussed, namely the Long Short Term Memory and LightGBM

A. Long Short Term Memory (LSTM)

In this research, one of the considered methods is the LSTM. This network is a recurrent neural network that utilized memory in order to predict the outcome. Unlike the standard RNN, the LSTM provides a solution to the vanishing gradient in RNN. [4]Common utilization of LSTM is in the field of Natural Language Processing (NLP), where work vocabularies were stored in order to predict texts and queries.

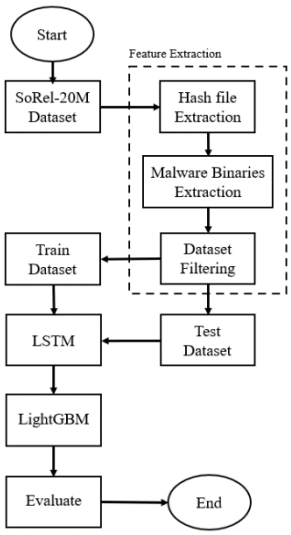

Figure 1: General System Architecture

LSTM is a feedforward neural network that utilizes the sequence of its data to predict the outcome. One of the main differences between the LSTM and RNN is the gated cell memory lock of LSTM. As seen in figure 2, the LSTM cell gate utilizes a combination of sigmoid functions and nd tanh activation functions. After each activation function comes to the gates, which control the flow of memory in each cell. Each gate functions as the input gate, the output gate, the cell state, and the forget gate. [5] To further understand the LSTM, the equation of an LSTM cell can be seen as follows:

$$f_i = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_i = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\underline{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \underline{C_t} \quad (4)$$

$$o_i = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

Where: gates are identified as *i* the input gate, o the output gate, and f the forget gate. The bias is denoted as b and the weight of the matrices is W. $\underline{C}$ is the Candidate state, while C is the new state. The h, x, and t were the output, input, and input time. Lastly, the σ represent the sigmoid function. [6][4]

There were 2 hidden layers and 256 neurons utilized in this LSTM model, where ReLu was used as the activation function. In addition to that is the sigmoid activation function of another layer of 1 neuron.

B. LightGBM

To reduce the computational time of learning, Light GBM will be implemented in complimentary to LSTM. Light GBM is a vertical tree gradient as it

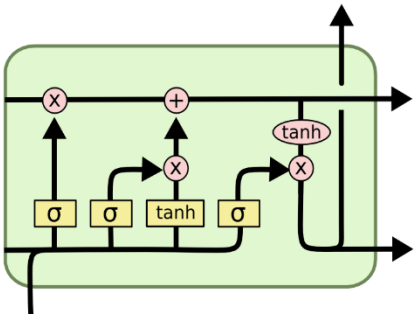

Figure 21. LSTM Gated Cell

produces high-speed results, is lighter on resources, and is high in accuracy. Although despite this it is important to note that LightGBM is not suited for small datasets as it is prone to overfitting.[7]

LightGBM utilizes a feature parallel technique that is different from the traditional feature parallel approach. In lightGBM, all data components know how to split the data, as compared to vertical splitting where it takes a huge amount of time. In this method, the data component finds the local best fit and communicates to its local component to find the performance best fit, and then it will communicate. The local best fits can be defined as thresholds, features and etc.[8]

The equation for calculation of the split test in LightGBM can be seen in equation 7. Gradient-based One Side Sampling (GOSS) was used by Light GBM to compute for information gain. In GOSS the larger and the smaller trailing gradient were used to be consistent with the information gain.

$$\tilde{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_r^j(d)}\right) \quad (7)$$

Where: x is the training set of n numbers, where every x is a vector of size s in space X^s. the negative gradient of the loss function set is g. A and B were the larger and the smaller gradient

In traditional data-parallel, all of the histograms of the data were compared, which is composed of all local histograms merged to form a global histogram. For the data parallel of Light GBM, Reduced Scatter is used to merge and compare histograms from different non-overlapping data components. From the local histograms, the data components find their local best fit, which will, later on, be synced with the global histogram to find the best fit. Subtraction is utilized by the LightGBM to reduce the processing time. With this logic, using the tree-based algorithm as a reference, communication is only done by 1 leaf and obtain its neighbor's histogram to subtract also.

IV. EXPERIMENTAL DATA AND ANALYSIS

In the experimentation, the SoReL-20M dataset was utilized, which is a 20 million Sophos - ReversingLabs dataset. The SoReL-20M datasets were unarmed PE malware. The proposed method extracts features of the PE malware and performs a train split test on the classifier. The classifier is a combination of LSTM and LightGBM. The confusion matrix score analysis yielded an overall accuracy of 91.73%.

After introducing the LightGBM, the image from figure 2. c was yielded and confusion matrix analysis. Comparing the two confusion matrix the processing time is reduced by 5% with the benefit of lighter resources. The advantage that this method proposes is its robustness compared to its predecessor, which used RNN.

V. CONCLUSION

This research proposes a non-signature-based malware detection using the SoReL-20M Dataset. The LightGBM provides a lighter load when it comes to resources. On the other hand, LSTM provides a more dynamic analysis of the datasets. The experimentation result yielded an accuracy of 91.73% with the aid of the proposed method.

[1] S. S. Tirumala, M. R. Valluri, and D. Nanadigam, "Evaluation of Feature and Signature based Training Approaches for Malware Classification using Autoencoders," *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. doi: 10.1109/comsnets48256.2020.9027373.

[2] "A learning model to detect maliciousness of portable executable using integrated feature set," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 2, pp. 252–265, Apr. 2019.

[3] S. A. Roseline, S. Abijah Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm," *IEEE Access*, vol. 8. pp. 206303–206324, 2020. doi: 10.1109/access.2020.3036491.

[4] M. A. I. Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. 2020. doi: 10.1109/niles50944.2020.9257950.

[5] A. Akandeh and F. M. Salem, "Slim LSTM NETWORKS: LSTM_6 and LSTM_C6," *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2019. doi: 10.1109/mwscas.2019.8884912.

[6] Y. Li and Y. Lu, "LSTM-BA: DDoS Detection Approach Combining LSTM and Bayes," *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*. 2019. doi:

10.1109/cbd.2019.00041.
[7] R. H. A. E. Rudd, “SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection,” *Cornell University*, Dec. 14, 2020. https://arxiv.org/abs/2012.07634 (accessed Sep. 04, 2021).
[8] Y. Zhang, X. Zhao, and Z. Li, “Facilitated and Enhanced Human Activity Recognition via Semi-supervised LightGBM,” *2020 IEEE Globecom Workshops (GC Wkshps*. 2020. doi: 10.1109/gcwkshps50303.2020.9367452.