Comparing Deep Learning Based Malware Detection Techniques in Cloud IaaS

McDole, Andrew D.[1] , Jeffrey Kimmell[1] , Maanak Gupta[1] , and Mahmoud Abdelsalam[2]
[1]Tennessee Technological University, Cookeville, TN, USA
[2]Manhattan College, Riverdale, NY, USA

BACKGROUND

Cloud infrastructure has presented great opportunities for consumers to harness virtualized computing power on demand, but it also present a major attack vector for attackers.

Because of this, constant monitoring of assets is essential for datacenters to protect their customers data and privacy. One solution to this monitoring is machine learning or more specifically, deep learning which is usually in the form of neural networks which imitate brain structure to learn.

STATIC ANALYSIS

In static malware analysis, files are scanned for malicious content before they are executed on a system. Once a program is deemed to be benign, it is allowed to execute normally without further monitoring. This approach often fails in cloud infrastructures where attackers can inject malware into existing applications which are already deemed to be safe and not subject to rescanning.

DYNAMIC ANALYSIS

Dynamic Analysis, as opposed to static analysis, attempts to analyze already running applications for signs of malicious activity. Certain metrics from a running virtual machine and its processes can be used to categorize that virtual machine as being infected with malware or running normally.

A neural network can take in this metric information and learn how to predict when a virtual machine is infected or not.

By combining machine learning and dynamic analysis techniques, accurate identification of malware infected virtual machines can be achieved

PREVIOUS WORK

Initial work has been conducted in this area [1][4]. This paper delivers a methodology to extract and format information from virtual machines in a cloud infrastructure and format this information for processing in a 2-Dimensional Convolutional Neural Network (2D CNN).

$$\mathbf{X}_t = \begin{bmatrix} & f_1 & f_2 & \cdots & f_n \\ p_1 & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_m & \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$

2D matrix of features and processes for input into a 2D CNN [4]

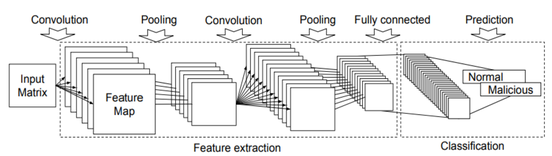
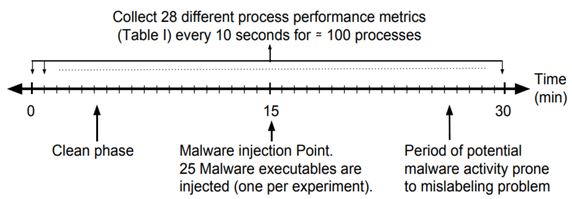


Diagram of a 2D CNN [4]

[1] used Openstack [2] to virtualize a 3-tier web architecture and simulated web requests using a multi-process traffic generator. Experiments were 30 minutes long where a single malware was injected into one virtual machine after 15 minutes. Results were collected every 10-seconds.



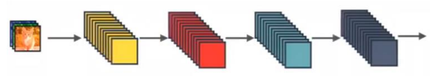
Data Collection Overview [4]

This model resulted in a ~75% accuracy when tested with different batch sizes. A 3-dimensional CNN has been used [1] to increase the accuracy to ~97%.

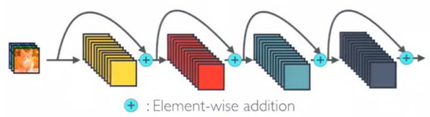
OUR RESEARCH

The model used in [1] is an older neural network model and newer models have been developed that may provide more accuracy. Some modern convolutional neural networks include:

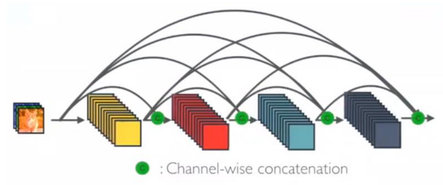
- Inception (GoogLeNet)
- ResNet (Residual Network)
- ResNetXt
- DenseNet



Standard CNN Concept [3]



ResNet Concept [3]



DenseNet Concept [3]

Results

Our dataset was composed of 113 data collection experiments which were broken up into training data (60%), validation data (20%), and testing data (%20). Each model was trained for 100 epochs and the best parameters were chosen by periodically testing the model against the validation set after each epoch. The following is a table depicting the results for each model.

Table 1. Results for Different Evaluation Metrics

Model	Accuracy	Precision	Recall	F1
LeNet-5	89.2	94.7	80.9	87.2
ResNet-50	88.4	86.0	88.9	87.4
ResNet-101	86.6	82.3	89.7	85.9
ResNet-152	89.5	89.0	87.8	88.4
DenseNet-121	92.9	100	84.6	91.5
DenseNet-169	92.8	99.7	84.4	91.4
DenseNet-201	92.8	99.5	84.6	91.5

It is important to point out that while some models may be more accurate than others, it is wise to take other considerations into account such as training time and detection time.

These metrics should be utilized when selecting a model for a real world use case. The table below shows the accuracy, training time and detection time for each model.

Table 2. Training and Detection Time

Model	Validation Accuracy	Epoch Reached	Elapsed Time (s)	Detection Time (ms)
LeNet-5	89.9	29	170	54
ResNet-50	90.7	67	1815	96
ResNet-101	87.0	60	2940	130
ResNet-152	88.7	99	7029	165
DenseNet-121	92.1	32	1683	164
DenseNet-169	91.9	81	5848	209
DenseNet-201	91.5	36	3060	249

Future Work

Other neural networks such as Recurrent Neural Networks (RNN) also show potential in regard to malware detection capabilities. RNN's temporal dynamic behavior could allow these models to predict performance metrics to a more accurate degree as well as perform more efficiently since the model will be analyzing the time relationship between the inputs.

References

[1] M. Abdelsalem, R. Krishnan, and Y. Huang, "Malware Detection in Cloud Infrastructures using Convolutional Neural Networks," in *Proc. of the 2018 IEEE 11th International Conference on Cloud Computing, July 2-7 2018, San Francisco, CA* [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed: Nov. 27 2019].

[2] D. Deloche, *Unfolded basic recurrent neural network [Online Image].* Available: Wikipedia.org, [Accessed: Nov. 27 2019]

[3] S. Tsang, *Standard CNN, ResNet, and Densenet Concepts [Online Image].* Available: Wikipedia.org, [Accessed: Nov. 27 2019]

[4] Abdelsalam M., Krishnan R., Sandhu R. (2019) *Online Malware Detection in Cloud Auto-scaling Systems Using Shallow Convolutional Neural Networks.* In: *Foley S. (eds) Data and Applications Security and Privacy XXXIII. DBSec 2019*. Lecture Notes in Computer Science, vol 11559. Springer, Cham