# Analyzing Machine Learning Approaches for Online Malware Detection in Cloud

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

## BACKGROUND

Cloud computing has seen an explosion of growth in recent years. Developments within cloud computing have led to companies such as Google and Amazon to offer Infrastructure as a Service (IaaS) platforms that allow consumers to utilize virtual machines in order to offer services to other consumers. The extensive growth of IaaS has been accompanied with numerous security concerns. With the possibility of devastating cascading affects, detecting malware that could disrupt IaaS providers has become a top priority for security researchers.

## DYNAMIC MALWARE ANALYSIS

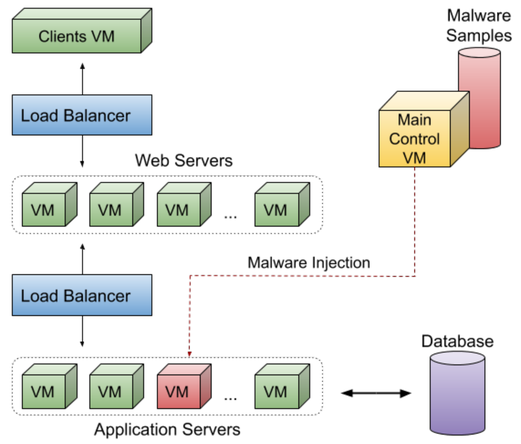
Dynamic malware analysis is performed by analyzing an application that is already running, typically in a sandbox environment. The metrics that are gathered from this application can be used to determine if the application is behaving maliciously or not. While dynamic analysis is preferred over static analysis, which only scans for known signatures of malicious applications, sophisticated malware can detect if it is being run in a sandbox environment and cease malicious activity until it reaches the actual machine.

## ONLINE ANALYSIS

Online malware detection, which is our approach in this work, is done by analyzing the system performance metrics of the actual machine as opposed to the process metrics of the running applications. This approach continuously monitors the behavior of the machine and assumes that malware will eventually find its way onto the machine. It also prevents malware from bypassing detection measures.

## RELATED WORK

Developing malware detection methods has become an issue that numerous researchers have attempted to tackle. Our work is unique due to our approach of using system level performance metrics as well as being specific to cloud environments. Works such as [1] [2] propose methods that utilize other features such as API calls and do not analyze the performance of their methods in cloud specific environments.



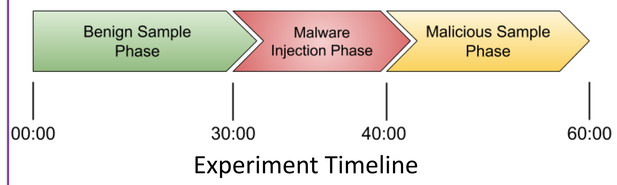

Experiment Deployment

## SETUP

Our experimental set up utilized Openstack to set up a testbed that used traffic to emulate real world cloud behavior, this set up is introduced in previous works [3] [4]. The testbed consisted of a control node that was responsible for the dashboard, storage, network, identity, and computing. The control node was accompanied by computes nodes that were strictly used for computing purposes. In order to create as realistic scenario as possible, we deployed a commonly used three tier web architecture to deploy our experiments. Malware types we injected include:

- DOS
- Backdoor
- Trojan
- Virus

Our experiments were run for a total of one hour. We conducted 113 experiments, one for each of our malware samples.




Experiment Timeline

## MODELS USED

We analyze numerous machine learning methods in order to determine which method is best for our use case. The methods we analyze are:

- Convolutional Neural Network (CNN)
- Support Vector Machine (SVM)
- Random Forest Classifier (RFC)
- K-Nearest Neighbor (KNN)
- Gradient Boosted Classifier (GBC)
- Gaussian Naive Bayes (GNB)

## RESULTS

Our results show that the CNN model is the best approach for our use case. The CNN model was able to achieve the highest accuracy and F1 scores. The F1 score is perhaps the most important metric because it signifies the balance between precision and recall. The CNN model also produced no false positives, signified by the 100% precision score, which is also an extremely important factor to consider in a real world use case.

DETAILED PERFORMANCE RESULTS FOR THE DIFFERENT ML MODELS

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| CNN | 92.9% | 100% | 84.6% | 91.5% |
| SVC | 87.56% | 86.2% | 80.91% | 83.47% |
| RFC | 89.36% | 99.71% | 72.80% | 84.15% |
| KNN | 72.34% | 66.6% | 57.67% | 61.81% |
| GBC | 81.47% | 75.22% | 77.87% | 76.57% |
| GNB | 58.09% | 48.06% | 98.57% | 64.61% |

This table illustrates the total time to train each model as well as the total time it took each model to detect a malicious sample. These metrics also play an important role in determining which method is right for a specific use case. Even though the CNN model takes the longest to train, its far superior performance justifies this time required for training.

TIME COST FOR THE MODELS

| Model | Time to Train (s) | Detection Time (ms) |
|---|---|---|
| CNN | 1683 | 164 |
| SVC | 989 | 11 |
| RFC | 20 | 3900 |
| KNN | 28 | 118 |
| GBC | 167 | 40 |
| GNB | 2 | .9 |

## References

[1] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), vol. 2. IEEE, 2016, pp. 577–582.

[2] Y. Fan, Y. Ye, and L. Chen, "Malicious sequential pattern mining for automatic malware detection," Expert Systems with Applications , vol. 52, pp. 16–25, 2016.

[3] 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].

[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