ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 12, December 2019

Comparative Analysis of LSTM Sequence-Sequence and Auto Encoder for real-time anomaly detection using system call sequences

Jayesh Soni[1], Nagarajan Prabakar[2], Himanshu Upadhyay [3]

Ph.D. Student, School of Computing and Information Sciences, Florida International University, Miami, USA[1]

Associate Professor, School of Computing and Information Sciences, Florida International University, Miami, USA[2]

Sr. Research Scientist, Applied Research Center, Florida International University, Miami, USA[3]

ABSTRACT: Designing a robust anomaly detection system for a computer system and applications, is very essential for system security. Host-based Intrusion Detection System monitors system call sequences to prevent the execution of malicious codes on the host. This study applies and compares deep learning based LSTM (Long Short Term Memory) Sequence-to-Sequence (Seq-Seq) and AutoEncoder in detecting an anomaly in Windows system call sequences.We use memory forensic techniques to extract system call sequences data of benign and malicious processes.Subsequently, we pre-process the extracted system call sequences to obtain valid system call sequences through filtering and ordering of redundant system calls. Our comparison analysis shows that LSTM Seq-Seq with a sequence length of ten gives high anomaly detection rate of 97.2% as compared to autoencoder.

KEYWORDS: Anomaly Detection; LSTM; AutoEncoder; Deep Learning; SystemCall

I. Introduction

Computer systems that work in a networked environment are vulnerable to different types of malicious activities. An intrusion detection system monitors activities on a network or system to detect malicious signs. Network-based and host-based are two different types of detection systems. Monitoring single system activity is host-based intrusion detection systems whereas monitoring host-to-host communication is network-based intrusion detection systems. Anomaly-based and Signature-based are also types of intrusion detection systems. The anomaly-based techniques compare the perceived behaviors against normal behaviors constructed from prior knowledge and if observed behavior deviates largely than it is declared as an attack. The signature-based approaches match the perceived behaviors against a database of known malicious patterns. The anomaly-based methods can possibly detect previously unseen attacks.

Recurrent neural network (RNN) based deep learning models yield extraordinary performance in tasks such as economic forecasting, bioinformatics, signal processing, and other tasks that consider temporal modeling.System-call sequences can be considered as a language used for communication between systems and users. Based on this, system calls correspond to words and system-call sequences correspond to sentences to detect anomalous system-call sequences. In this paper, we compare LSTM (Long Short Term Memory) Sequence to Sequence and AutoEncoder for detecting an anomaly in a process by using system call sequence.

In the next section, we discuss the state of the art techniques. The data collection procedure is described in Section III. Next, we provide an in-depth overview of the proposed anomaly detection algorithms in section IV. The results are discussed in section V. Finally, we conclude with section VI.

II. Related work

Traditional Intrusion Detection Systems compare the behavior of software with a database collection of known attacks. When a pattern of behavior matches the one in the database, the behavior is considered anomalous. In this paper, we focus on intrusion detection systems that work by creating a learning based model trained on the behavior of a normal process and then monitors the system operations for deviations from the normal data. These systems are called anomaly detection systems.

Forrest et al. proposed STIDE [1]which takes normal traces and creates multiple sequences, each with sliding window length of *n* and stores it in a database. Furthermore, sequence data of unknown behavior of length *n* were also extracted and compared with the database. If an unknown sequence is detected in a trace then it is considered as anomalous. Warrender et al. [2] identify the anomalous attack by counting the number of mismatches in STIDE output during the last *m* calls.

Machine learningbased techniques such asself-organizing maps neural network [3], multilayer perceptron [4]have been researched to detect anomaly in a process[5]. For static analysis, classification algorithms such as decision tree, SVM, and clustering algorithms such as K-means are used widely. Several deep learning based algorithm have been developed for anomaly detection that requires both the benign and malicious data. Such algorithm works as binary classifier[6][7][8]. Tandon[9] developed a learning algorithm that learns the system calls arguments and its sequences.
In this paper, we compare two unsupervised anomaly detection techniquesthat learn temporal behaviors of system call sequences. Our approach utilizes a technique that trains on normal data and checks for abnormal activities that differ from normal behavior. Such procedures identify unknown or malicious attacks on the system.

III. Data Collection

The proposed framework shown in Figure 1 uses the Libvirt library to manage virtual machines running on the Xen hypervisor. We extract and process the system call information by inspecting the VM process through the Introspector and Security Agent modules. We use Google's rekall profile and LibVMI library to extract system call features. Introspector and Security Agent modules, written in Go Language are used to process the request and extract the system call traces from VM with LibVMI functions. The Libvirt library allows us to create, start, and stop virtual machines of Windows.

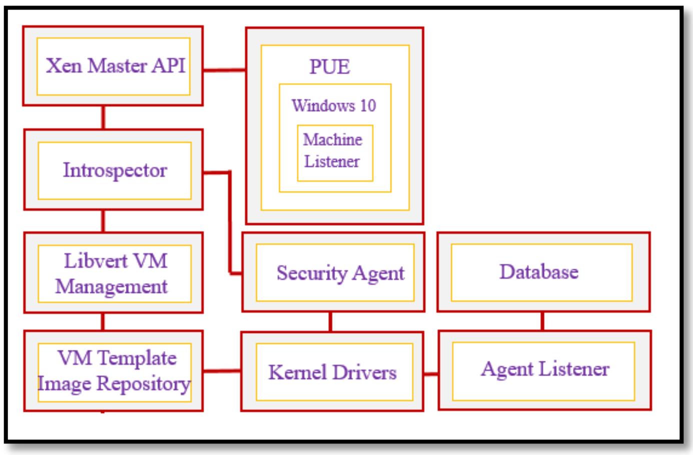

Fig.1. Data Collection Framework

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 12, December 2019

IV. Detection Algorithm

In this section, we discuss an overview of anomaly detection algorithms, namely LSTM Seq-Seq and Autoencoder. These algorithms have low run-time computational complexity which is crucial for anomaly detection systems. They work well even for large training examples with high dimensions. We trained both the algorithms without providing any training labels considering them as unsupervised learning.

A: *Long Short Term Memory (LSTM) Seq-Seq*

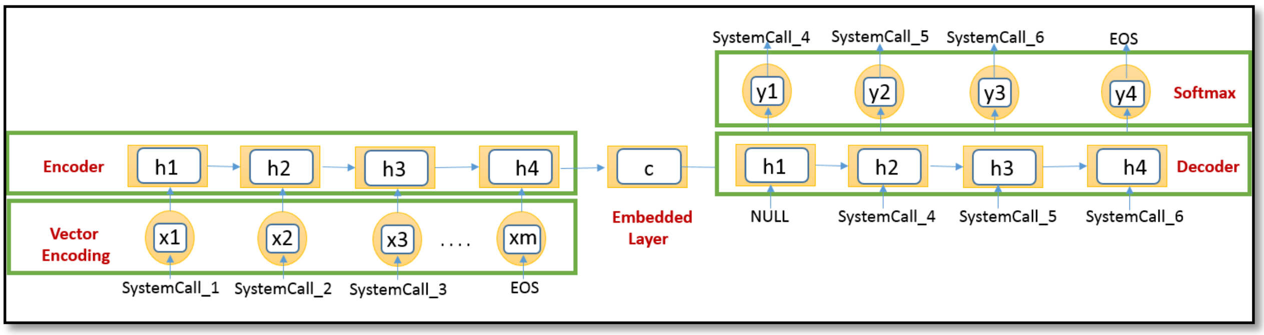

Fig. 2. LSTM Seq-Seq Architecture

LSTM Seq-Seq has three major components: encoder model, embedded layer and decoder model as depicted in Figure 2. Each system-call is transformed to one-hot encoding and fed to the encoder model. At the embedded layer, each incoming system calls gets multiplied by matrix W to continuous space. Decoder model learns the next sequence of system calls. LSTM internal state is updated recurrently at the hidden layer at each time step. A softmax activation function is used at the output layer, to estimate the probability values of the possible system calls in the sequence, P $(X_i | X_{1:i-1})$.

With the chain rule, the probability of the sequence of the system call can be estimated. We train this LSTM Seq-Seq based system call model using benign (normal) system call sequence data. The training condition maximizes the likelihood of the system call sequence and minimizes the cross-entropy loss. We trained the LSTM Seq-Seq algorithmwith 140 K system call sequences with varying sequence lengths. Given a new query system call sequence, on the assumption that malicious system call sequence behavior deviates from learned benign behavior, a sequence with an average negative log-likelihood above a specified threshold is considered as a malicious sequence, while a sequence with an average negative log-likelihood below the threshold is classified as normal.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 12, December 2019

B: AutoEncoder

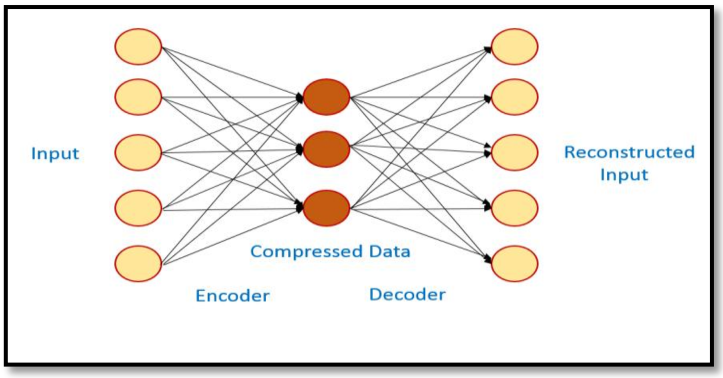

Fig. 3. AutoEncoder Model

Figure 3 shows general AutoEncoder model. The reconstruction error is the fundamental metric in autoencoder to detect anomalies. During the learning phase, an autoencoder tries to minimize this error. In doing so, it learns the hidden relationships between the features of the input data. An AutoEncoder has an input layer as an encoder, hidden layer which contains the compressed data and output layer as the decoder. During training, it takes the data, compresses it to lower dimension and reconstructs the data with a compressed dimension. If we feed a trained autoencoder with testing data, it should reproduce the new input similar to the training data, if the new test data fed as the input resemble the training data. The reconstruction error is high if the test input data deviates from the train data and flag it as malicious. All autoencoders have the same structure. Our AutoEncoder model comprises of four layers: I) an input layer with the number of neurons equals to the number of features, II) a densely connected repeat vector layer with number of neurons to be of window size, III) a dense output layer with linear activations and IV) a time distributed dense layer. We create this network after experimenting with different hyperparameter optimization and topologies. In summary, the following steps are followed: 1) create an autoencoder with the above network; 2) train the autoencoder using system call data generated through a hypervisor during normal operating conditions; 3) Use the reconstruction error to identify anomalies in new data.

V. Results

A. *Result Analysis of LSTM Seq-Seq Model*

We obtained system call sequences data through the Xen hypervisor and implemented LSTM Seq-Seq architecture to detect anomalies by considering the temporal ordering of system calls. Following hyperparameters are tuned to arrive at the optimized LSTM model:
Sequence length: It is the length of the last processed system call sequence, used as input to predict theadjacent next set of system call sequences.
Epochs: Number of iterations, a model passes through the entire data for training.
Batch-size: Number of system call sequences passed to the LSTM network in a single iteration.
To evaluate the LSTM model, we define its accuracy as:
Accuracy = (Number of correct system call sequence predictions) / (Total number of system call sequence predictions performed)
We trained the LSTM model with Adam optimizer and categorical cross-entropy as a loss metric. We found that the batch size of 256 gives higher accuracy.
Figure 4 shows the accuracy and loss with the LSTM Seq-Seq model for different sequence lengths with batch size as 256 and epochs as 100. We found that the model trained with a sequence length of ten has the lowest loss and highest

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 12, December 2019

accuracy. Our trained model gives 97.2% accuracy with a loss of 0.08. Table 1 shows the time taken to train the LSTM model per epoch for varying sequence lengths.

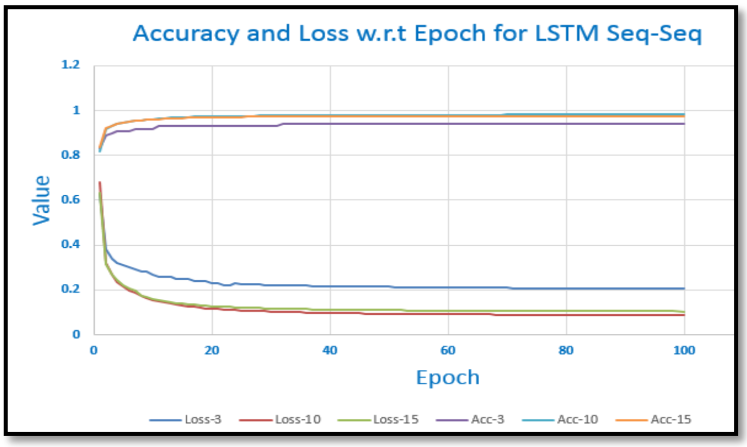

Fig. 4. Loss and Accuracy w.r.t Epoch for LSTM Seq-Seq with varying Sequence Length

Table 1. Time per Epoch for LSTM Seq-Seq with varying Sequence Length

Sequence Length	*Three*	*Ten*	*Fifteen*
Time in msec for each Epoch	180	370	530

B. *Result analysis of AutoEncoder*

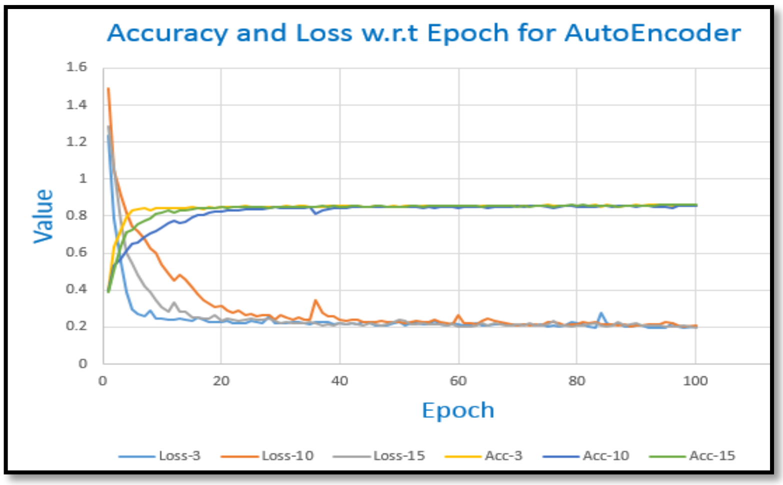

Fig. 5. Loss and Accuracy w.r.t Epoch for AutoEncoder with varying Sequence Length

Similarly, we trained the AutoEncoder model with Adam optimizer with linear activation function and categorical cross-entropy as a loss metric.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 12, December 2019

To evaluate the AutoEncoder model, we define its accuracy as:
Accuracy = (Number of correct system call sequence predicted with zero reconstruction error) / (Total number of system call sequence predicted)
Figure 5 shows the accuracy and loss with the AutoEncoder model for different sequence lengths with batch size as 256 and epochs as 100. We found that the model trained with a sequence length of ten has the lowest loss and highest accuracy. AutoEncoder model gives the highest accuracy at 86% accuracy with a sequence length of ten and a loss of 0.21. Table 2 shows the time taken to train the AutoEncoder model per epoch for varying sequence lengths.

Table 2. Time per Epoch for AutoEncoder with varying Sequence Length

Sequence Length	*Three*	*Ten*	*Fifteen*
Time in msec for each Epoch	230	460	610

VI. Conclusion

In this study, we implemented and performed an in-depth comparative analysis of LSTM Seq-Seq and AutoEncoder for anomaly detection techniques using system-call sequences. Through introspection, we extracted the system-call sequence of the benign and malicious processes. Through filtering and ordering the redundant system calls, we obtained the valid sequences. The experimental results demonstrate that LSTM Seq-Seq with a sequence length of ten has the best performance for anomaly detection, reaching the accuracy of 97.2% as compared to AutoEncoder. We further show that the LSTM Seq-Seq has low training time compared to the AutoEncoder model.

VII. Acknowledgment

This research was supported by the Department of Defense (DOD)-Test Resource Management Center, USA.

References

1. Forrest, S., Hofmeyr, S. A., Somayaji, A., And Longstaff, T. A. 1996b. A sense of self for unix processes. In Proceedings of the IEEE ISRSP. 120–128.
2. Forrest, S., Warrender, C., And Pearlmutter, B. 1999. Detecting intrusions using system calls: Alternate data models. In Proceedings of the IEEE ISRSP. IEEE Computer Society, 133–145.
3. Lotfi Shahreza M, Moazzami D, Moshiri B, Delavar MR (2011) Anomaly detection using a self-organizing map and particle swarm optimization. Sci Iran 18(6):1460
4. Xuan Dau Hoang, Jiankun Hu, and Peter Bertok. A multi-layer model for anomaly intrusion detection using program sequences of system calls. In Proc. 11th IEEE Intl. Conf. Citeseer, 2003.
5. Soni, J., Prabakar, N. and Kim, J-H. (2017) " Prediction of Component Failures of Telepresence Robot with Temporal Data ". 30th Florida Conference on Recent Advances in Robotics.
6. J. Soni, N. Prabakar, and H. Upadhyay, "Deep Learning approach to detect malicious attacks at a system level," in WiSec'19: Proceedings of 12th ACM Conference on Security & Privacy in Wireless and Mobile Networks, May 15-17, 2019, Miami, FL, USA. 2 pages.
7. J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," Advances in neural information processing systems, pp. 943–949, 1998.
8. J. Soni, and N. Prabakar, " Effective Machine Learning Approach to Detect Groups of Fake Reviewers," Proceedings of the 14th International Conference on Data Science (ICDATA'18), Las Vegas, NV, 2018.
9. G. Tandon and P. Chan. Learning rules from system call arguments and sequences for anomaly detection. In ICDM Workshop on Data Mining for Computer Security (DMSEC), pages 20–29, 2003.

BIOGRAPHY

Mr. Jayesh Soni is a Ph.D. student in computer science department at Florida International University. His main research focus is in the field of artificial intelligence and cyber security. Mr. Jayesh is also a Teaching Assistant in the Machine Learning course and has done few publications and 1 book chapter.