LSTM for Anomaly-Based Network Intrusion Detection

Sara A. Althubiti
Department of Computer Science
North Carolina A&T State University
Greensboro, NC
saalthub@aggies.ncat.edu

Eric Marcell Jones Jr.
Department of Computer Science
North Carolina A&T State University
Greensboro, NC
emjones1@aggies.ncat.edu

Kaushik Roy
Department of Computer Science
North Carolina A&T State University
Greensboro, NC
kroy@ncat.edu

Abstract— **Due to the massive amount of the network traffic, attackers have a great chance to cause a huge damage to the network system or its users. Intrusion detection plays an important role in ensuring security for the system by detecting the attacks and the malicious activities. In this paper, we utilize CIDDS dataset and apply a deep learning approach, Long-Short-Term Memory (LSTM), to implement intrusion detection system. This research achieves a reasonable accuracy of 0.85.**

Keywords—intrusion detection system, anomaly detection, long short-term memory

I. INTRODUCTION

Web applications have been widely used and are often under the threat of attack due to rapid development of the internet and its extensive usage. Cyber attackers might cause damages to systems in different means, for instance, disrupting access, stealing data, or destroying a specific target. Web Intrusion Detection Systems (WIDSs) are developed to detect abnormal activities and behaviors and activities in a system to diminish expected risks of attacks. Intrusion detection system (IDS), an influential approach, is primarily implemented to detect abnormal activities in a target applications or computers. An IDS has two main methods [1], signature-based detection and anomaly-based detection. Signature-based detection is used to detect identified attacks using rule-based methods. Conversely, anomaly-based detection is utilized to detect both the known and unknown attacks by learning their behavior.

Researchers studied the effectiveness of intrusion detection techniques in [2, 3]. Flow-based CIDDS-001 was analyzed using different machine learning algorithms, including *k*-nearest neighbor and *k*-means clustering [4]. Also, this dataset was evaluated among different datasets using deep neural network to classify attacks in IoT networks [5]. Recently, deep learning-based approaches perform relatively well than the traditional machine learning approaches [6]. In this research, we implement a Long Short-Term Memory (LSTM) [6] model for intrusion detection and evaluate the performance of the LSTM model on the CIDDS-001 dataset. Then we compile the model utilizing an "rmsprop" optimizer, seeking an optimal solution for the multi-class intrusion classification problem using accuracy rate as a performance measure.

In [4], CIDDS-001 dataset was analyzed from machine learning perspective. In [5], deep neural network (DNN) was applied on CIDDS-001 dataset. DNN performance was evaluated utilizing different validation methods such as cross-validation, recurrent cross-validation, and subsampling on different datasets. Authors applied a grid search to explore the best DNN learning parameters for each dataset individually. DNN performance was certainly satisfactory in the attack detection in wireless environment. In [7], an IDS model with deep learning was built through employing a Long Short-Term Memory-Recurrent Neural Network (LSTM-RNNS), and the IDS model was trained and assessed utilizing the KDD Cup 1999 dataset. A deep learning method was approved as an effective technique to implement IDSs and discover the best suitable optimizer among six optimizers (RMSprop, Adagrad, Adadelta, Adam, Adamax, and Nadam) for LSTM RNN [8]. In [9], an RNN-IDS was projected as a deep learning method to investigate the performance of binary and multi-class model classifications on NSL-KDD dataset. Also, researchers in [9] investigated the effects of different number of neurons and learning rates of the RNN-IDS model. Moreover, the proposed method was compared with other techniques, such as Support Vector Machine (SVM), J48, Random Forest (RF), naïve Bayes, and Artificial Neural Network (ANN). Results showed that the RNN-IDS was a suitable classification model with higher accuracy and lower false acceptance rate, and performed reasonably well for both binary and multi-class classification methods as compared to conventional approaches.

Researchers in [10] applied Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs) for intrusion detection to estimate the time series model of the KDDCup99 dataset. Confusion matrix,

accuracy, Receiver Operating Characteristic (ROC) curve and the corresponding Area-Under-the-Curve (AUC) were used to evaluate the performance. Trained LSTM neural networks learned all the five traffic classes moderately (normal, denial of service, network probes, user-to-root, remote-to-local attacks).

II. LSTM FOR ANOMALY DETECTION

Recurrent Neural Network (RNN) is a well-known model for deep learning [8]. RNN is employed to recognize generated image and text and interpret machine with high performance. However, RNN fails to capture long-term dependency that connects consecutive tasks and the base of vanishing gradient descent. LSTM is anticipated to capture long-term dependencies [11]. The primary goal of LSTM is to achieve disappearing gradient descent which is an optimization algorithm to find artificial neural networks weights to avoid long-term dependency problems. This research uses the LSTM to detect anomalies.

Keras [12], a high-level neural networks API, written in Python and capable of running on top of TensorFlow, is used for LSTM implementation. We conducted experiments by finding hyper-parameter values to attain best performance measures of IDS. The trained LSTM model utilize an input layer with ten neurons which corresponds to the ten features, a hidden layer with six neurons, and an output layer with five neurons. The number of iterations was set to 200 epochs. The network weights, used in this research, range from 0 to 0.05. During training stage, we specify the loss function to evaluate the weights. In this experiment, we use the logarithmic loss, which is defined in Keras as "c**ategorical_crossentropy"** since this research aims to solve a multi-class classification problem. Loss function tends to assess the variation between predicted and actual values. **fit()** function is used to fit the model on the data and specifies epochs and batch size. After training the model on the complete dataset, we can assess accuracy and model loss as the performance measures of the same data.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this effort, we use CIDDS-001 dataset, first reported in [13], for performance evaluation of LSTM model. It is a labelled flow-based dataset for anomaly-based IDS evaluation. We describe the dataset in the following section. The data of network traffic was obtained from an external server and an OpenStack environment. Additionally, the original data of CIDDS-001 comprises 13 features: *Src IP, Src Port, Dest IP, Dest Port, Proto, Date first seen, Duration, Bytes, Packets, Flags, Class, AttackType, and AttackID*. The network attacks are categorized into five classes: normal, suspicious, unknown, attacker, and victim. In this study, we only used the external traffic and we omitted three features, which are *AttackID, AttackType*, and *AttackDescription* because they provide information about the nature of attacks and are not considered in this study. The external data traffic comprises 671241 flows.

There are 4 numeric features and 6 nonnumeric features in the CIDDS-001 external server traffic dataset. Because the input value of LSTM should be a numeric matrix, we must convert all nonnumeric features, such as 'protocol_type', 'IP Address of the source node', 'IP Address of the destination node' and 'flag' features, into numeric form.

In this study, we divided the external data into two sets: training and testing. 67% of the data was used as the training set and the rest was for considered for testing set. Moreover, the training set contains 449,731 instances and the testing set comprises 221,510 instances. The hyperparameters are optimized in an attempt to improve the training and test accuracies. Initially, the LSTM model use the hyperparameter values listed in Table I. The learning rate was adjusted and updated to particular weight at the end of each batch, and the best results were achieved with the value of 0.01. The single hidden layer makes the model deeper and precise. The number of epochs is adjusted for the entire training set. Batch size is number of patterns exposed to the network prior to the updated weights. We apply LSTM RNN model with "*rmsprop*" optimizer using parameters as reported in Table II. This optimizer is suitable for large datasets and efficient calculation.

TABLE I. THE INITIAL VALUES OF HYPERPARAMETERS OF THE MODEL

Hyperparameter name	Value
Learning rate	0.01
Hidden layers	6
Number of epochs	200
Batch size	500

TABLE II. THE OPTIMIZER VALUES OF THE MODEL

Rmsprop optimizer	Value
Learning rate	0.01
Rho	0.05
Epsilon	1e-8
Decay	0.0

TABLE III. THE COMPARISON WITH OTHER ALGORITHMS

Classifier	Precision	Recall	FPR	Accuracy
LSTM	0.8514	0.8834	0.1722	0.8483
SVM	0.8043	0.8322	0.1698	0.7942
Naïve Bayes	0.7925	0.8299	0.1325	0.7756
MLP	0.8372	0.8523	0.1734	0.8124

Testing the designated optimizer for LSTM model in an attempt to detect the intrusion is a vital step. Initially, the learning rate for the *rmsporop* optimizer is 0.01, which achieves the best accuracy. For the training phase, model reached an accuracy of 0.8713 for a loss function of 0.3810 as shown in Fig. 1 and Fig. 2. The classification performance of the LSTM model was

evaluated using accuracy. The accuracy plot in Fig. 1 indicates that both the datasets are trained well and obtained satisfactory performances. In the Fig. 2, the model maintains comparable performance for both the training and test data. For evaluation objective, we compare our LSTM result to other classifier algorithms using some metrics such as Accuracy, Precision, Recall, False Positive Rate (FPR). Nevertheless, the FPR for LSTM is a little higher than two other algorithms, but percentages of the precision, recall and accuracy are the best comparing to the three other algorithms as shown table III.

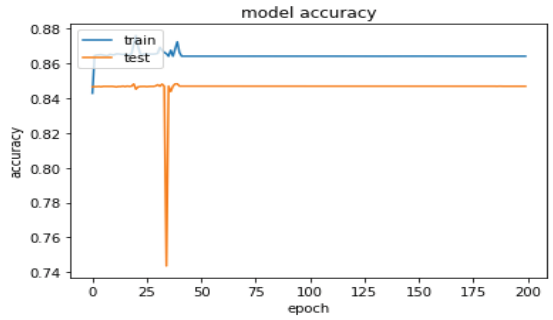

Fig. 1. Model accuracy

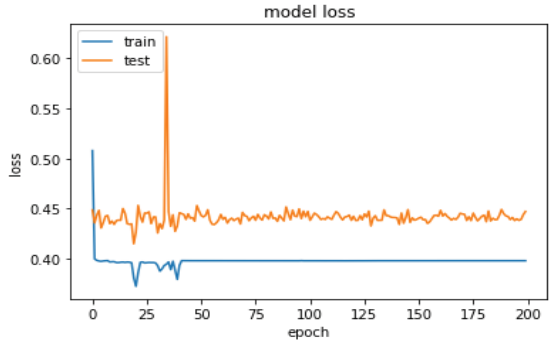

Fig. 2. Model loss

IV. CONCLUSION

In this study, we found rmsprop optimizer is suitable for LSTM model in intrusion detection. Using rmsprop for LSTM model using rmsprop optimizer can construct a competent multi-class classifier for IDS. The LSTM model obtained a reasonable accuracy of 0.8483. Also, we found that LSTM performs better than SVM, MLP, and Naïve Bayes techniques for a multi-classification problem. For future work, we will apply LSTM on CIDDS-001 datasets and assess the performance of LSTM with other optimizers. We also plan to compare the performances of LSTM with the traditional classifiers.

ACKNOWLEDGMENT

This research is based upon work supported by the Science & Technology Center: Bio/Computational Evolution in Action Consortium (BEACON) and NSF REU grant (#1460864).

REFERENCES

[1] Mell, Peter, Vincent Hu, Richard Lippmann, Josh Haines, and Marc Zissman. "An overview of issues in testing intrusion detection systems." (2003).

[2] Auxilia, Michael, and D. Tamilselvan. "Anomaly detection using negative security model in web application." In *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, pp. 481-486. IEEE, 2010.

[3] Valeur, Fredrik, Darren Mutz, and Giovanni Vigna. "A learning-based approach to the detection of SQL attacks." In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 123-140. Springer, Berlin, Heidelberg, 2005.

[4] Verma, Abhishek, and Virender Ranga. "Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning." *Procedia Computer Science* 125 (2018): 709-716.

[5] Tama, Bayu Adhi, and Kyung-Hyune Rhee. "Attack Classification Analysis of IoT Network via Deep Learning Approach."

[6] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

[7] Kim, Jihyun, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim. "Long short term memory recurrent neural network classifier for intrusion detection." In *Platform Technology and Service (PlatCon), 2016 International Conference on*, pp. 1-5. IEEE, 2016.

[8] Kim, Jihyun, and Howon Kim. "An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization." In *Platform Technology and Service (PlatCon), 2017 International Conference on*, pp. 1-6. IEEE, 2017.

[9] Yin, Chuanlong, Yuefei Zhu, Jinlong Fei, and Xinzheng He. "A deep learning approach for intrusion detection using recurrent neural networks." *IEEE Access* 5 (2017): 21954-21961.

[10] Staudemeyer, Ralf C., and Christian W. Omlin. "Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data." In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pp. 218-224. ACM, 2013.

[11] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

[12] Keras: A high-level neural networks API, Available at: Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow.

[13] Ring, Markus, Sarah Wunderlich, Dominik Grüdl, Dieter Landes, and Andreas Hotho. "Flow-based benchmark data sets for intrusion detection." In *Proceedings of the in Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS). 1em plus 0.5 em minus*, pp. 361-369. 2017.