ISSN 2278-3091

Volume 9, No.4, July – August 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse340942020.pdf

https://doi.org/10.30534/ijatcse/2020/340942020

Network anomaly detection technique based on LSTM network and UNSW-NB15 dataset

Lai Van Duong [1], Tisenko Victor Nikolaevich[2], Pham Thi Thuong[3], Dong Xuan Anh[5], Nguyen The Lam[5]

[1,3,4,5]Information Assurance dept. FPT University, Hanoi, Vietnam, duonglvse05009@fpt.edu.vn,
ThuongPTSE05856@fpt.edu.vn, AnhDXSE06086@fpt.edu.vn, lamntse63326@fpt.edu.vn,
[2]Department Quality Systems, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg,
Polytechnicheskaya, 29, v_tisenko@mail.ru

ABSTRACT

Cyber-attack is a dangerous attack technique used by attackers to attack information systems. There are two methods to detect cyber-attacks that are based on the sign set and based on abnormal behavior of the network data. In particular, the detection method based on abnormal behavior analysis techniques is being highly effective due to strong changes in technologies and algorithms in machine learning and deep learning. In this paper, we propose a method for detecting anomalies in networks using the Long Short Term Memory (LSTM) network and the UNSW-NB15 dataset Accordingly, based on the UNSW-NB15 dataset that was analyzed and extracted features of the network traffic, we use the LSTM network to classify into abnormal behavior or normal behavior. To see the effectiveness of the LSTM network in the problem of classifying and evaluating network behavior, in this paper, we will change the structure and parameters of the algorithm. These experimental results are presented in the experimental section of the paper.

Key words: Anomaly detection, Machine learning, Network traffic, Long Short Term Memory.

1. INTRODUCTION

The cyber-attack is a form of dangerous attack that has increased rapidly in both the number of recorded attacks and the extent of their damage to organizations and businesses. The studies [1], [2] classified cyber-attack techniques into two main methods: Passive Attack and Active Attack. According to the reports [3], [4], in 2019, the cyber attack techniques are considered to be one of the most dangerous attack techniques. Currently, there are two main methods for detecting cyber-attacks: signature-based method through the rule sets, and anomaly-based method based on data analysis and statistics to find out abnormal characteristics in the network [1], [2].The anomaly-based method is usually based on two main techniques that are machine learning and deep learning to classify abnormal and normal behavior [1], [2]. In this paper, we propose a method of detecting cyber-attack using the deep learning method based on the LSTM network. LSTM network is currently one of the most widely used deep learning networks by its ability to store and remember learned data. LSTM network has been applied in many different fields, but in the problem of detecting anomalies in the network, there are not many studies mentioned. Besides, the studies [1], [2] listed and analyzed some data sets commonly used for cyber-attack detection such as DARPA/KDD Cup99, CAIDA, NSL-KDD, ISCX 2012, UNSW-NB15, etc.In these datasets, the UNSW-NB15 data set is built and developed relatively in accordance with real network systems [1], [5]. Therefore, in this paper, we will use the UNSW-NB15 dataset to experiment cyber-attack detection methods.

2. RELATED WORKS

The study [6] proposed using the LSTM network for abnormal detection in the network. In their study, the author used KDD 99 dataset as experimental data for the LSTM network.

The authors expose, in the paper [7], a new idea of Deep Learning, which associates Auto-Encoder and Deep Belief Network (DBN). The Auto-Encoder was used for the purpose of decreasing the dimensionality of data and identifying the principal features of it, while the DBN had the mission of detecting the dubious code. The test of the new suggestion of model was done with the dataset KDD Cup 99, the assessment of results was compared with only a single DBN. The attainment has announced that the new method is completely more accurate with less consumption of time. However, the authors did not specify why they preferred to combine DBN and Auto-Encoder to form this hybrid.

In the study [8], Vikash Kumar et al. proposed a method to classify cyber-attack techniques based on UNSW-NB15 by using different rule sets. NourMoustafa et al. [9] proposed the Geometric Area Analysis Technique to detect cyber-attacks by using Trapezoidal Area Estimation. To evaluate the effectiveness of the proposed method, the authors conducted experiments on UNSW-NB15 and NSL-KDD datasets. Experimental results in the study showed the superiority of the UNSW-NB15 dataset over the NSL-KDD dataset. Besides, research [10] presents a scalable framework for building an effective, lightweight anomaly detection system based on two well-known datasets, the NSL-KDD and UNSW-NB15. This framework includes three modules:

capturing and logging, pre-processing, and a new statistical decision engine called the Dirichlet mixture model based on anomaly detection technique. The first module sniffs and collects network data while the second module analyses and filters these data to improve the performance of the decision engine. Finally, the decision engine is designed based on the Dirichlet mixture model with a lower upper interquartile range as a decision engine. SikhaBagui et al. [11] proposed the cyber-attacks detection method based on Naïve Bayes, and Decision Trees (J48) algorithm. In their experimental section, the research team [11] used these algorithms in turn to classify different cyber-attack components in the UNSW-NB15 dataset. In the study [12], the authors proposed a model to detect cyber-attacks using stacking techniques. Accordingly, in the training process of their model, the author uses machine learning algorithms consisting of K-Nearest Neighbors (KNN), Decision Tree (DT), and Logistic Regression (LR) in order to build a model based on the UNSW- NB15 and UGR'16 datasets. The study [13] evaluated the effectiveness of 8 machine learning algorithms (consisting of 2-layer and 3-layer algorithms) for network intrusion detection. This is a good idea, but it requires the use of the Microsoft Azure Machine Learning Studio system to apply in practice. In this research, we proceeded to distinguish between attack and normal based on pure machine learning algorithms and the use of Apache Spark technology. Our results are similar to the results of the method that authors [13] proposed, but our performance and experimental configuration are much simpler than the research [13].

3. The method of detecting anomalies in the network using the LSTM algorithm and the UNSW-NB15 dataset

3.1. Description of the UNSW-NB15 dataset

3.1.1. Data collection method

The UNSW - NB15 dataset was built by using the IXIA PerfectStorm tool to extract the mixture of attack operations in the network [5]. Over 100 GB of raw network traffic was captured by the tcpdump tool and processed via Argus engine, Bro-IDS, and twelve algorithms written in C # to extract 49 features that will be presented in the next section. These features save in CSV format.

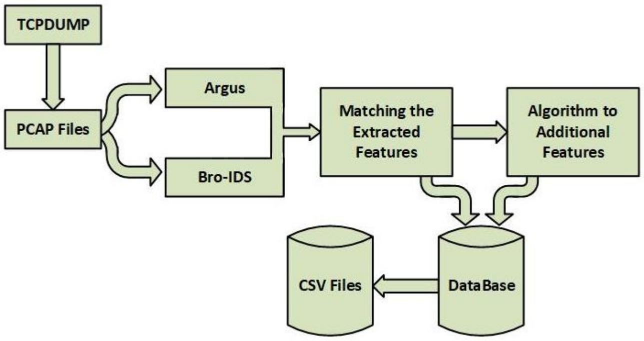
TCPDUMP
PCAP Files
Argus
Bro-IDS
Matching the Extracted Features
Algorithm to Additional Features
CSV Files
DataBase

Fig. 1. Collection and processing model for building UNSW - NB15 dataset

- Argus is a tool to process raw network packets (pcap file) and aggregate to features of the flow. This tool consists of Argus-server to write packages to Argus file with the binary form and Argus-client to extract features.
- Bro-IDS is a tool to analyze open-source network traffic, helps to check all traffic in order to against malicious operation. This tool is configured to form three types of log files from the pcap file: conn file records all information about connections in pcap file, HTTP file records all HTTP connections, and FTP file record all operation of the FTP service.

Table 1 below shows detailed statistics about the dataset including total flow, the number of records according to the network protocol, the number of normal records and abnormal records, and the number of source/destination IP addresses.

Table 1: Statistics of the components of the UNSW - NB15 dataset

		Number of flows
Total		2540047
Protocol	TCP	1495074
	UDP	990435
	ICMP	524
	Others	54014
Label	Normal	2218764
	Abnormal	321283
Unique	Src IP	43
	Dest IP	47

3.1.2. Description of selected features

The selected features are divided into six groups:

- Flow features: include features used to identify network flow such as IP address, port number, and protocol.
- Basic features: include connection description features.
- Content features: consist of features of TCP/IP protocol, and features of HTTP application layer protocol.
- Time features: include time-related features such as packet arrival time, start/end time and round trip time of TCP protocol.
- Additional generated features. Features in this group can be divided into two smaller groups: general purpose features and connection features.
- Labelled features: are labels for records.

Details of the features of each group are described in detail in Table 2.

Table 2: The list of features in UNSW - NB15 dataset

No.	Name	Type	Description
1. Flow features			
1	srcip	nominal	Source IP address
2	sport	integer	Source port number
3	dstip	nominal	Destination IP address
4	dsport	integer	Destination port number
5	proto	nominal	Transaction protocol
2. Basic features			
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
9	dbytes	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
3. Content features			
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the flow packet size transmitted by the src
24	dmeansz	integer	Mean of the flow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.

4. Time features			
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of ’synack’ and ’ackdat’.
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
5. Additional generated features			
37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No. of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No. of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No. of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No. of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
6. Labelled features			
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

3.2.The LSTM algorithm

LSTM network was first introduced in the documents [14], [15],and then improved and disseminated. LSTM works effectively on many different problems. It is designed to avoid the loss of information when solving the problem of long-term dependencies. The characteristic of LSTM is the ability to remember information for long periods of time without the need to train to do so. The LSTM network has a chain structure using four interacting layers as shown in Figure 2.

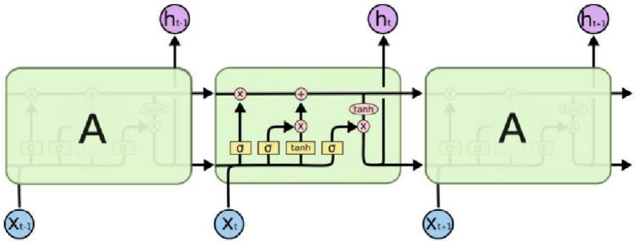

Figure 2:. Structure of LSTM Networks

The neuron layers used in LSTM include *tanh*and *sigmoid*. The most special characteristic of LSTM is the cell state which is the upper line of the diagram in Figure 2. The cell state runs through all the network nodes and only has a little linear interaction, so information can be easily transmitted over the network without being changed. The LSTM has the ability to add or remove information for the cell state. The addition or deletion of information is adjusted through the gates. The gates filter the information through it and determine the amount of information that is passed into the cell state. Each gate consists of a sigmoid neural net layer (symbol σ in the rectangle) and a pointwise multiplication operation. The sigmoid layer has the output in the segment [0, 1]. If the output of the sigmoid is 0, no information is passed. If the output is 1, all information is passed. The LSTM network uses three gates with this structure.

The first step in the processing of LSTM is to decide what information is to be removed in the cell state. The decision would be made by the sigmoid layer that is also called the "forget gate layer". The input of the gate is the result h_{t-1} of the previous step and the data x_t to compute. Then, the output is a number in the segment [0, 1] for each cell state value C_{t-1}. The next step is to decide which new information is saved to the cell state. Firstly, a layer of sigmoid called the "input gate layer" is used to decide how much information is updated into the cell state. Besides, a *tanh* layer return a vector of new candidate values $\widetilde{C}_t$, that could be added to the cell state. The cell status update is done by multiplying the output of first step by the old value C_{t-1} of cell state, then adding with the result of multiplying $\widetilde{C}_t$ by the output of the "input gate layer". Finally, to calculate the output value, the *tanh* function is used to push the values of the cell state in [-1, 1]. The result of sigmoid gate is then multiplied by the value of the cell state that is passed through the*tanh*function to give the output value.

4. EXPERIMENTS AND EVALUATIONS

4.1. Evaluation criteria

In this paper, we specify that the abnormal record is labeled as ***positive,*** and normal records are labeled as ***negative.*** The metrics used to evaluate the effectiveness of the abnormal detection method in our paper include:

- True positive (TP) is the number of abnormal records that are correctly predicted.
- False positive (FP) is the number of normal records that are incorrectly predicted as abnormal.
- True negative (TN) is the number of normal records that are correctly predicted.
- False negative (FN) is the number of abnormal records that are incorrectly predicted as normal.
- Accuracy: the ratio between the number of points correctly predicted and the total number of points in the test dataset.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\ \%$$

- Precision: the ratio of the number of true positive points among those classified as positive (TP + FP). High Precision value means that the accuracy of the found points is high.

$$precision = \frac{TP}{TP + FP} \times 100\ \%$$

- Recall is defined as the ratio of the number of true positive points among those that are actually positive (TP + FN). High Recall value means that the TPR (true positive rate) is high meaning that the rate of missing the actual positive points is low.

$$\mathrm{Re}\ call = \frac{TP}{TP + FN} \times 100\ \%$$

- F1-score is the harmonic mean of precision and recall. The higher F1, the more effective the classifier is.

$$F1 = \frac{2*Precision*Recall}{Precision+Recall}$$

4.2. Experimental results

Based on the experimental dataset consisting of 2,218,764 clean records and 321,283 attack records [5], we randomly divided into 2 parts including 80% for training and 20% for testing. Table 3 below shows the experimental result of anomaly detection in network based on the LSTM network.

Table 3: Experiment results of LSTM network with changing network parameter

N. of LSTM	Acc	Pre	Rec	F1
1	0.912640	0.873944	0.806087	0.838646
2	0.914865	0.873393	0.816005	0.843724
3	0.940917	0.875058	0.821883	0.847637

With the results in Table 3, it can be easily seen that the more complex the network architecture is, the more number of hidden layers and the corresponding number of nodes, the better the learning capability of the model is and the more accurate the test results are. With simple models, classifiers give Accuracy, Precision, Recall, F1 measures as 91.3%, 87.4%, 80.61%, 83.86% respectively. In the most complex model, the Accuracy, Precision, Recall, F1 measures are respectively 94.1%, 87.5%, 82.2%, 84.8%. This is a relatively good result for the classifier when the dataset has a huge discrepancy between clean data and malicious data.

5. CONCLUSION

Cyber-attack techniques have always been and will always be major challenges for intrusion monitoring and detection systems. With the goal of testing and evaluating the LSTM network in anomalies detection, in the study we proposed some models that apply the LSTM network and UNSW - NB15 dataset. The experimental results in Table 3 show that the LSTM network has the capability of detecting anomalies in the network well and steadily even when the dataset has a huge difference between normal data and abnormal data. Besides, in order to see the effectiveness of the LSTM network, we attempted to experiment, evaluate and optimize parameters of the LSTM network in order to seek the model that gives the best results for the task of detecting cyber-attack. Finally, the experimental results in the paper proved the suitability of the deep learning method in general and the LSTM algorithm in particular in the anomalies detection problem

REFERENCES

[1] R. Markus., et al., **A survey of network-based intrusion detection data sets.***Computers & security*, vol. 8, no. 6, pp. 147–167, 2019.

[2] Kh. Ansam. et al., **Survey of intrusion detection systems: techniques, datasets and challenges**,**Cybersecurity**, vol. 20, pp. 2-20, 2019. https://doi.org/10.1186/s42400-019-0038-7

[3] Imperva, **2019 Cyberthreat Defense Report**, 2019. [Online]. Available: https://www.imperva.com/resources/reports/CyberEdge-2019-CDR-Report-v1.1.pdf [Accessed 1 April 2020].

[4] **Sophos,** "Sophos 2020 Threat Report," [Online]. Available: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-uncut-2020-threat-report.pdf [Accessed 11 April 2020].

[5] **The UNSW-NB15 Dataset Description**, "University of New South Wales Canberra," 2020. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ [Accessed 14 April 2020].

[6] AlaeddineBoukhalfa, AbderrahimAbdellaoui, Nabil Hmina, HabibaChaoui. **LSTM deep learning method for network intrusion detection system.** *International Journal of Electrical and Computer Engineering (IJECE*) Vol. 10, No. 3, June 2020, pp. 3315-3322.

[7] Li Y., Ma R. and Jiao R., **A Hybrid Malicious Code Detection Method Based on Deep Learning**, *International Journal of Security and Its Applications (IJSIA)*, vol. 9, no. 5, pp. 205–216, 2015. https://doi.org/10.14257/ijsia.2015.9.5.21

[8] K. Vikash., et al., **An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset**,vol 22, *Cluster Computing.* doi: 10.1007/s10586-019-03008-x, 2019.

[9] N. Moustafa., et al**., Novel Geometric Area Analysis Technique for Anomaly Detection using Trapezoidal Area Estimation on Large-scale Networks**, *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 2332-7790, 2017. https://doi.org/10.1109/TBDATA.2017.2715166

[10] N. Moustafa et al., **Big Data Analytics for Intrusion Detection System**: Statistical Decision-Making Using Finite Dirichlet Mixture Models," doi :10.1007/978-3-319-59439-2_5, 2017.

[11] S. Bagui, et al., **Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset**, *Security and Privacy*, doi: 10.1002/spy2.91, 2019.

[12] S. Rajagopal., et al., **A predictive model for network intrusion detection using stacking approach**,*International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 10, No. 3, pp. 2734-2741, June 2020.

[13] S. Rajagopal, et al., **Performance analysis of binary and multiclass models using azure machine learning**, *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 10, No. 1, pp. 978-986. https://doi.org/10.11591/ijece.v10i1.pp978-986

[14] Sepp Hochreiter, Jürgen Schmidhuber. **Long Short-Term Memory.** *Neural Computation*9(8):1735 - 1780, 1997.

[15] Alex Sherstinsky. **Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network.** *Physica D: Nonlinear Phenomena*. Vol404, pp. 1-48. 2020. https://doi.org/10.1016/j.physd.2019.132306