Optimization of Cyber-Attack Detection Using the Deep Learning Network

Lai Van Duong [1†],

Information Assurance dept. FPT University, Hanoi, Vietnam

Summary
Detecting cyber-attacks using machine learning or deep learning is being studied and applied widely in network intrusion detection systems. We noticed that the application of deep learning algorithms yielded many good results. However, because each deep learning model has different architecture and characteristics with certain advantages and disadvantages, so those deep learning models are only suitable for specific datasets or features. In this paper, in order to optimize the process of detecting cyber-attacks, we propose the idea of building a new deep learning network model based on the association and combination of individual deep learning models. In particular, based on the architecture of 2 deep learning models: Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), we combine them into a combined deep learning network for detecting cyber-attacks based on network traffic. The experimental results in Section IV.D have demonstrated that our proposal using the CNN-LSTM deep learning model for detecting cyber-attacks based on network traffic is completely correct because the results of this model are much better than some individual deep learning models on all measures.

Key words: ***cyber attack, combined deep learning; abnormal behaviors of cyber-attacks; detection attacks***

1. Introduction

1.1 The problem

Four main cyber-attack methods that have been identified by the research [1] are: Fabrications, Interceptions, Interruptions, and Modifications. These attack methods perform various techniques to attempt to conceal and hide themselves from intrusion detection and prevention systems. In order to identify abnormal behaviors of cyber-attack techniques, studies [2, 3, 4, 5, 6, 7, 8, 9, 10] used the network traffic datasets with different feature sets such as KDD 99, DARPA/KDD Cup99, CAIDA, NSL-KDD, ISCX 2012, UNSW-NB15, IDS 2018. However, because attack techniques are getting more sophisticated, the features and behaviors defined from previous datasets will not give highly effective due to unsuitability with actual data. Based on the analysis in [1], we think that the UNSW-NB15 dataset may be suitable for the architecture and characteristics of the current network, so it is suitable for attack campaigns in reality. The studies [1, 11, 12, 13] listed a number of approaches for cyber-attack detection including the approach using rule sets, the approach using behavior profiles, and the approach using a combination of rule and behavior. In particular, the cyber-attacks detection approach using behavior analysis techniques has brought high efficiency due to the support of machine learning and deep learning models. We noticed that: due to the structure and characteristics of deep learning methods, each model has certain advantages and disadvantages. Therefore, the selection of deep learning algorithms and models for experimental datasets plays a decisive role in the results of cyber-attack detection. Therefore, in this paper, we propose a new cyber-attack detection method based on combining many different deep learning models. Our purpose is to combine many different deep learning models in order to take advantage of their advantages and minimize the remaining disadvantages. Specifically, in our study, we propose a CNN-LSTM combined deep learning model to detect abnormal behaviors of cyber-attacks based on the UNSW-NB15 dataset.

1.2 Contributions of Paper

The practical and scientific significance of our paper includes:

- Proposing an approach to combine individual deep learning networks into a synchronous deep learning network. In the proposal, we use deep learning networks in serial, so that the output of one network will be the input of another. With this approach, we try to combine individual deep learning networks and take advantage of them for processing and computation to find the signs and behaviors of cyber-attacks.
- Propose architecture of some CNN-LSTM combined deep learning models based on individual deep learning networks LSTM, CNN. These are new combined deep learning models, have not been applied by any research and proposal in the problem of detecting cyber-attacks based on Network traffic. To evaluate the effectiveness of the proposed model, we compare and evaluate the proposed deep learning model with individual models. During the experiment, we conduct evaluations to select the most optimal parameters and the most optimal combined deep learning models for the task of detecting cyber-attacks..

Manuscript received July 5, 2021
Manuscript revised July 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.7.19

2. Related Works

In the study [14], Vikash Kumar et al. proposed a method for classifying cyber-attack techniques based on UNSW-NB15 using rulesets. Nour Moustafa et al. [15] proposed Geometric Area Analysis Technique for cyber-attack detection using Trapezoidal Area Estimation. To evaluate the effectiveness of the proposed method, the authors conducted experiments on the UNSW-NB15 and NSL-KDD datasets. The experimental results in this study showed the superiority of the UNSW-NB15 dataset compared to the NSL-KDD dataset. Besides, the study [16] presented a scalable framework for building an effective and lightweight anomaly detection system based on two well-known datasets, the NSL-KDD and UNSW-NB15. Sikha Bagui et al. proposed in their study [17] a method to detect cyber-attacks based on the Naïve Bayes and Decision Tree (J48) machine learning algorithms. In their experimental section, the research team [17] used these algorithms in turn to classify different cyber-attack components in the UNSW-NB15 dataset. In the study [18], Cho et al. proposed two tasks: detecting cyber-attacks using machine learning algorithms and optimizing features using algorithms such as IG, PCA. Experimental results showed that the team's proposals were relatively good. However, because feature optimization algorithms have large computational times and high complexity, a large calculation system is required. In the study [19], Zhao et al. proposed a botnet detection method based on analyzing abnormal behaviors of traffic and flow. Besides, the approach to detect botnet and cyber-attack using the CTU 13 dataset was proposed by Chowdhury et al. [20]. In addition, Ahmed [21] proposed using the ANN deep learning algorithm to classify abnormal connections. Besides, Cho et al. [22, 23, 24] proposed a method to detect cyber-attacks based on network traffic using machine learning and deep learning algorithms. Specifically, in the study [23], the authors propose a deep learning model that combines Bidirectional Long Short-Term Memory (BiLSTM) and Graph Convolutional Networks (GCN) to analyze network traffic to detect cyber-attacks. Besides, in the studies [24, 25], the authors also proposed the CNN-LSTM method for detecting cyber-attacks based on the IDS 2018 dataset. Jiang et al. [26] proposed a deep learning model combining CNN with Recurrent Neural Networks (RNN) for anomaly detection in the intrusion detection and prevention system. In addition, there are also some approaches using other deep learning models such as MLP [27], LSTM [28].

3. Proposing the Detection Method

3.1. Introduction to the dataset:

The UNSW - NB15 dataset was built by using the IXIA PerfectStorm tool to extract the mixture of attack operations in the network [29]. 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.

- 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.

- These features save in CSV format. Table I 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

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 CNN-LSTM deep learning model for cyber-attack detection

3.2.1. Introduction to CNN

In deep learning, CNN is a class of deep neural network, most commonly applied to analyzing visual imagery. CNN is widely applied to solve problems such as image classification [30], object detection [31], segmentation [32, 33], face recognition [34], as well as applications in text classification [35], modeling sentences [36]. The detailed structure of CNN as well as the terms (stride, padding, MaxPooling) are detailed in the articles [37, 38]. The activation function used is ReLU (1).

$$( ) \quad \max(0, \ ) \tag{1}$$

3.2.2. Introduction to LSTM

In the study [39], Hochreiter and Schmidhuber introduced the architecture and mathematical foundations of the LSTM network. The LSTM network is a neural network developed on the structure of RNN [40] to overcome some problems related to Gradient Exploding and Gradient Vanishing when the network is too long. Typically, the LSTM network as well as the RNN network are able to remember information from the previous state of the network, so that they can process time-series data or series data. Figure 1 describes in detail the structure of a basic memory cell in the LSTM network with 4 gates having different tasks.

The gates are used to control how much information from the previous cell could be add or erase. At each time t, we have a hidden state $\mathbf{h}_t$ and a cell state $\mathbf{c}_t$ with the basic mathematical formulas shown below:

The input gate to control how much data to write:

$$\mathbf{i}_t = \sigma(\mathbf{W}^{(i)}\mathbf{h}_{t-1} + \mathbf{U}^{(i)}\mathbf{x}_t + \mathbf{b}^{(i)}) \tag{2}$$

The forget gate to control how much data will be erased:

$$\mathbf{f}_t = \sigma(\mathbf{W}^{(f)}\mathbf{h}_{t-1} + \mathbf{U}^{(f)}\mathbf{x}_t + \mathbf{b}^{(f)}) \tag{3}$$

The output gate to control how much data will go through:

$$\mathbf{o}_t = \sigma(\mathbf{W}^{(o)}\mathbf{h}_{t-1} + \mathbf{U}^{(o)}\mathbf{x}_t + \mathbf{b}^{(o)}) \tag{4}$$

And the new memory cell to control what will be write:

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}^{(c)}\mathbf{h}_{t-1} + \mathbf{U}^{(c)}\mathbf{x}_t + \mathbf{b}^{(c)}) \tag{5}$$

And two cell:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \tag{6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t \tag{7}$$

Where: $\mathbf{W}$ is the weight matrix of each gate corresponding to the hidden state of the previous cell; $\mathbf{U}$ is the weight matrix of each gate corresponding to the input at time t; $\odot$ is the element-wise product operator.

Fig. 1. The architecture of a hidden cell of LSTM deep learning network