Optimization of Cyber-Attack Detection Using the Deep Learning Network

Lai Van Duong [1†],
DuongLVSE05009@fpt.edu.vn
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).

$$f(x) = \max(0, x) \tag{1}$$

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.

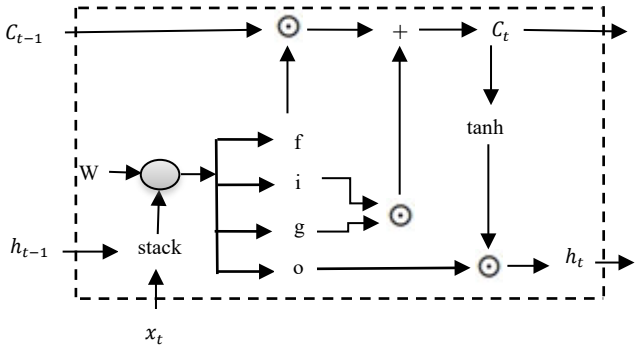

Fig. 1. The architecture of a hidden cell of LSTM deep learning network

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.

3.3 The proposed CNN-LSTM architecture

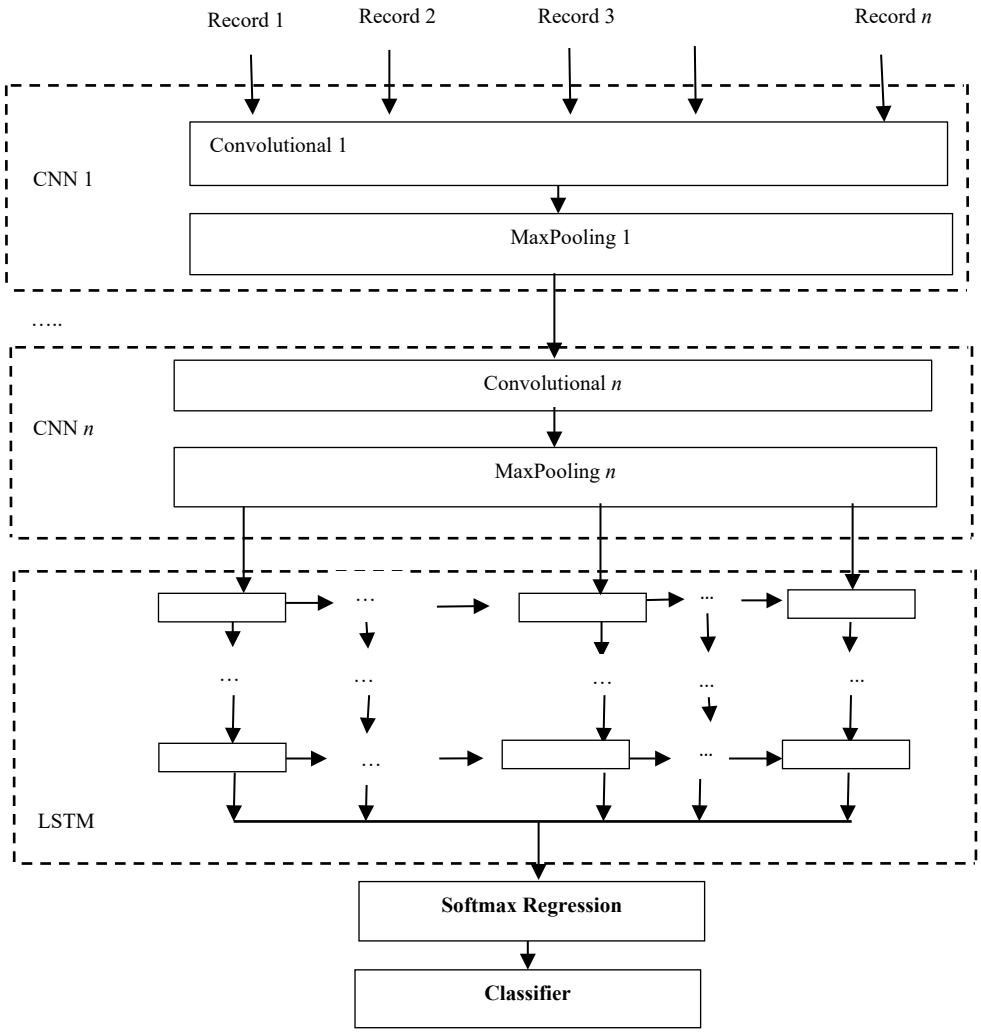

Fig. 2. Architecture of the CNN-LSTM deep learning network model

The procedure for detecting cyber-attacks has 3 steps:

- Step 1: Extract features of cyber-attacks using CNN: The cyber-attack records will be used as input to the CNN network (this CNN network consists of convolutional layers, max pooling, and ReLU activation function) to extract the characteristic of features that are adjacent to each other. The features extracted by CNN will be flattened into vectors and put into the LSTM network to continue to learn the patterns.
- Step 2: Aggregate abnormal features using LSTM: We will use the LSTM model with multiple hidden layers, modify the number of units, and simultaneously use regularization techniques such as drop out with different rates to find the best architectures. The LSTM network will extract the features of network flows depending on the flows sent before it. The output of the LSTM network is a vector containing the outstanding features of the flows which were extracted and selected by the CNN-LSTM model.
- Step 3: Classification: To classify into attack and normal, we use 2 layers: Fully Connected Layers and Softmax Layers. These classes have the following tasks:
  - **Fully Connected Layers:** This layer is like an MLP network that is responsible for learning the features processed through the CNN-LSTM layers. The detailed operating principle of Fully Connected Layers is as follows: MLP networks usually have 3 or more layers, with 1 input layer, 1 output layer and more than 1 hidden layer. The

formula for hidden layer is defined as formula (8) [41].

$$h_i^j = f(\sum_{k-1}^{n_{i-1}} w_{k,j}^{i-1} h_{i-1}^k) \quad i = 2,...,N \text{ and } j = 1,...,n_i \qquad (8)$$

In which, $w_{k,j}^i$ is the weight between neuron k in hiden layer i-1 and neuron j in hidden layer i and f is an activation function such as ReLU, sigmoid [41]. In this paper, the sigmoid function is used and described as formula (9).

$$f(x) = \frac{1}{1+e^{-x}} \qquad (9)$$

- **Softmax Layers:** This layer is responsible for calculating the probability of the output label. In which, the softmax function is described as below formula (10):

$$a_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad \forall i = 1,2,...,C \qquad (10)$$

Where: C is the number of classes; $z = [z_1, z_2, ..., z_C]$ is the output vector of the LSTM network corresponding to the input graph that needs to be classified; a_i is the probability that the input belongs to the ith class and calculated by the softmax function. Further note that the Softmax Regression function is only responsible for calculating the probability of belonging to classes of the input, but it is not meaningful in the feature extraction process. The highest probability of belonging to a class, that class will be assigned to the predicted record.

4. Experiments and evaluation

4.1. Experimental dataset

The experimental dataset in our paper includes 2,540,047 records [27] consisting of 2,218,764 normal records and 321,283 attack records. This dataset is randomly divided at the rate of 80% for training and 20% for testing.

4.2. Experimental scenario

To evaluate the effectiveness of the proposed model, we compare and evaluate according to the following scenarios:

- **Scenario 1:** compare the CNN-LSTM model with individual deep learning networks in other studies including MLP [27], CNN, LSTM [28].
- **Scenario 2:** compare and evaluate the CNN-LSTM model with another combined deep learning model, CNN-MLP.

4.3. Classification Measures

- **Accuracy:** the ratio between the number of correctly predicted points and the total number of points in the test dataset

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** the ratio between the number of true positive (TP) points and the number of points classified as positive (TP + FP).

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** is defined as the ratio between the number of true positive (TP) points and the number of points that are actually positive (TP + FN). A high recall value means a high true positive Rate (TPR) means that the rate of omission of positive points is low.

$$Recall = \frac{TP}{TP + FN}$$

Where: **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.

4.4. Experimental results

4.4.1. Experimental results of scenario 1

TABLE II. EXPERIMENTAL RESULTS OF CYBER-ATTACK DETECTION WITH SCENARIO 1

Deep Learning Model	N. of layers	N. of nodes / Dimension of hidden state	Evaluation			
			Acc	*Pre*	*Rec*	*F1*
MLP [27]	2	128-128	0.927	0.845	0.583	0.690
	3	128-256-256	0.937	0.908	0.606	0.727
	4	128-128-512-256	**0.943**	**0.819**	**0.760**	**0.8**
LSTM [28]	1	128	0.945	0.845	0.692	0.761
	2	128-256	0.951	0.878	0.711	0.786
	3	128-256-128	**0.962**	**0.898**	**0.789**	**0.840**
CNN	2	128-128	0.845	0.583	0.690	0.845
	3	128-256-256	0.908	0.606	0.727	0.908
	4	128-128-512-256	**0.819**	**0.760**	**0.8**	**0.819**

Based on the results in Table II, it can be easily seen that: with the MLP model, the more complex the network architecture is, the more hidden layers and the

corresponding number of nodes are, the better the learning ability of the model, and the more accurate test results are. With simple models, Accuracy is 92.7%, Precision is 84.5%, and Recall is 58.3%. In the model with the highest complexity, the Accuracy for the best classification is 94.3% which is a pretty good result. Overall, with the MLP model, the Precision achieved much higher results than Recall. One reason is the difference in the number of clean records and malicious records were too large leading to imbalanced classifications.

Regarding the LSTM model, we noticed that in the feature aggregation and extraction process, this model has the ability to memorize and learn the features, so it has significantly improved the classification results. Specifically, with the parameter LSTM [128 - 256 - 128], the LSTM model gave the best results on all measures. These results were higher than the results of the MLP model on all measures.

Regarding the CNN model, it is clear that this model was not as effective as the LSTM and MLP models. The reason is that the structure of the CNN network is often used in block-form processing problems, instead of the single-form as in this study. Besides, the results in Table II show that the CNN model has brought very good results for the process of accurately classifying cyber-attacks.

Comparing the classification results in Table II, seeing that the LSTM model gave better effect than the CNN and MLP models on measures such as Accuracy and Precision. However, the CNN network gave more accurate attack classification results than LSTM and MLP networks. This result shows that if you know how to combine LSTM and CNN networks, it will bring good results because of inheriting each other's advantages.

4.4.2. Experimental results of scenario 2

4.4.2.1. Experimental results of cyber-attack detection using the CNN-MLP model

TABLE III. EXPERIMENTAL RESULTS USING THE CNN-MLP MODEL

N. of CNN + MLP	Evaluation			
	Acc	*Pre*	*Rec*	*F1*
2-2	0.965	0.912	0.801	0.853
3-3	**0.968**	**0.907**	**0.836**	**0.870**
4-4	0.967	0.907	0.825	0.864

Table III shows the classification results of the CNN-MLP deep learning model. From Table III, seeing that the more complex architecture of the CNN-MLP model is, the higher the Accuracy is. However, the model achieved the best results on all measures in architecture 3CNN-3MLP. Comparing tables II, III, we noticed that the CNN-MLP model had a much better effect than the individual deep learning models. In particular, the CNN-MLP model gave the Accuracy about 3% higher and the Precision about 1% higher. As for Recall, there is a difference of about 4%. Obviously, CNN-MLP has shown remarkable efficiency because it has improved cyber-attack detection. Thus, it is clear that with the combination of CNN and MLP, the CNN-MLP model has greatly improved the efficiency of the cyber-attack detection process. The reason for this problem is this model takes full advantage of the advantages of hidden layers in the CNN network to process data. Not only that, the F1-score measure of the CNN-MLP model is also much better than other models. This shows a balance and stability between metrics of accurately detecting normal accesses and cyber-attack accesses.

4.4.2.2. Experimental results of cyber-attack detection using the CNN-LSTM model

TABLE IV. EXPERIMENTAL RESULTS USING THE CNN-LSTM MODEL

N. of CNN + LSTM	Evaluation of Flow			
	Acc	*Pre*	*Rec*	*F1*
2-2	0.973	0.913	0.869	0.890
3-3	0.979	0.948	0.882	0.914
4-4	**0.981**	**0.945**	**0.902**	**0.923**

After experimenting with many different models combining LTSM and CNN with fine-tuned parameters, Table IV is a summary evaluation of the models that gave the best results with the test dataset. We noticed that with the different number of parameters between the models, the experimental results of training and detecting cyber-attacks are also different. Specifically, the 2CNN-2LSTM model is the worst model on all measures. However, when increasing the number of hidden layers in each network, the CNN-LSTM model yielded the best results when using 4CNN-4LSTM with Accuracy of 98.1%, Precision of 94.5%, and Recall of 90.2 %. These results show that the CNN-LSTM model gave good results on both normal and attack detection.

Comparing the results in Tables II, III, IV, it is clear that our proposal of using the CNN-LSTM model in this study is more effective than the MLP [27], LSTM [28], and CNN-MLP models. This has proved that the CNN and LSTM networks have done very well in the feature extraction task: with only 2 to 4 CNN and LSTM networks can obtain the necessary features for classification. Adjacent CNN and LSTM layers are responsible for exploring more high-level features if any

5. Conclusion

In this paper, we have accomplished the purposes that are set out in our research proposal. In particular, with the approach using in serial many different deep learning

networks with complex computing and processing processes, we have succeeded in integrating individual deep learning networks according to the principle: the output of one deep learning model is the input of another. This helps to maximize the advantages of each deep learning network, thereby helping to improve the efficiency of the cyber-attack detection process. With this approach, we have provided intrusion detection systems with a new detection mechanism to effectively improve the ability to quickly and accurately detect threats from the internet. In addition, in this paper, based on the principles of building combined deep learning networks, we have succeeded in building a CNN-LSTM deep learning model based on two individual deep learning networks: CNN and LSTM. The experimental results in the paper have shown that the CNN-LSTM model has brought better results than the CNN, MLP, LSTM, CNN-MLP models on all measures. Besides, based on experimental scenarios, we have provided readers with criteria to select the parameters of each network in order to ensure the time and efficiency of the detection process. In the future, to improve the efficiency of cyber-attack detection, we will research and combine other deep learning networks to form new combined deep learning networks. In addition, we will also seek to connect deep learning networks with attention or inference layers to improve the ability to integrate and reproduce features of deep learning networks.

References

[1] Gilberto Fernandes Jr., Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi & Mario Lemes Proença Jr., "A comprehensive survey on network anomaly detection," Telecommunication Systems, vol. 70, pp. 447–489, 2019.

[2] Kamal Alieyan, Ammar Almomani, Ahmad Manasrah, Mohammed M. Kadhum, "A survey of botnet detection based on DNS," Neural Computing and Applications, vol. 28, pp. 1541–1558, 2017.

[3] Mohiuddin Ahmed, Abdun Naser Mahmood, Jiankun Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp 19-31, 2016.

[4] Sebastián García, Alejandro Zunino, Marcelo Campo, "Survey on network-based botnet detection methods," Security Comm. Networks, 2013. https://doi.org/10.1002/sec.800.

[5] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, Dan Garant, "Botnet detection based on traffic behavior analysis and flow intervals," Computers & Security, vol. 39, pp. 2-16.

[6] Manmeet Singh, Maninder Singh, Sanmeet Kaur, "Issues and challenges in DNS based botnet detection: A survey," Computers & Security, vol. 86, pp. 28-52, 2019.

[7] Monowar H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," IEEE Communications Surveys & Tutorials, vol. 16 (1), pp. 303–336, 2014.

[8] Sudipta Chowdhury , Mojtaba Khanzadeh , Ravi Akula, Fangyan Zhang, Song Zhang, Hugh Medal, Mohammad Marufuzzaman, Linkan Bian, "Botnet detection using graph based feature clustering," Big Data, vol. 4 (14), 2017. doi 10.1186/s40537-017-0074-7.

[9] Omar Y. Al-Jarrah, Omar Alhussein, Paul D. Sami Muhaidat, Kamal Taha, and Kwangjo Kim, "Data Randomization and Cluster-Based Partitioning for Botnet Intrusion Detection," IEEE Transactions on Cybernetics, vol 46 (8), pp. 1796 – 1806, 2016.

[10] Abdulghani Ali Ahmed, Waheb A. Jabbar, Ali Safaa Sadiq Hiran Patel, "Deep learning based classifcation model for botnet attack detection," Journal of Ambient Intelligence and Humanized Computing, https://doi.org/10.1007/s12652-020-01848-9.

[11] Sneha Kudugunta, Emilio Ferrara, "Deep Neural Networks for Bot Detection," arXiv:1802.04289v2.

[12] Samaneh Mahdavifar, Ali A. Ghorbani, "Application of deep learning to cybersecurity: A survey," Neurocomputing, vol. 347, pp. 149-176.

[13] Robert Luh, Stefan Marschalek, Manfred Kaiser, Helge Janicke, Sebastian Schrittwieser, "Semantics-aware detection of targeted attacks: a survey," J Comput Virol Hack Tech, vol. 13, pp. 47–85, 2017. doi 10.1007/s11416-016-0273-3

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

[15] 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.

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

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

[18] Cho Do Xuan, Hoang Thanh, Nguyen Tung Lam, "Optimization of network traffic anomaly detection using machine learning," International Journal of Electrical and Computer Engineering, vol. 11, no. 3, pp. 2360-2370, 2021.

[19] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, Dan Garant, "Botnet detection based on traffic behavior analysis and flow intervals," Computers & Security, vol. 39, pp. 2-16, 2013.

[20] Sudipta Chowdhury, Mojtaba Khanzadeh, Ravi Akula, Fangyan Zhang, Song Zhang, Hugh Medal, Mohammad Marufuzzaman & Linkan Bian, "Botnet detection using graph-based feature clustering," Journal of Big Data, vol. 4, no. 14, 2017.

[21] Abdulghani Ali Ahmed, Waheb A. Jabbar, Ali Safaa Sadiq, Hiran Patel, Journal of Ambient Intelligence and Humanized Computing, 2020. https://doi.org/10.1007/s12652-020-01848-9.

[22] Cho Do Xuan, Lai Van Duong, Tisenko Victor Nikolaevich, "Detecting C&C Server in the APT Attack based on Network Traffic using Machine Learning," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 11(5), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110504.

[23] Cho Do Xuan, Hoang Mai Dao, Hoa Dinh Nguyen, "APT attack detection based on flow network analysis techniques using deep learning," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 3, pp. 4785-4801, 2019.

[24] Cho Do Xuan, Hoang Mai Dao, "A novel approach for APT attack detection based on combined deep learning model," Neural Comput & Applic, 2021. https://doi.org/10.1007/s00521-021-05952-5

[25] P. Sun et al., "DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System," Security and Communication Networks, vol. 2020, pp. 1-11, 2020.

[26] F. Jiang et al., "Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security," in IEEE Transactions on Sustainable Computing, vol. 5, no. 2, pp. 204-212, 1 April-June 2020. https://doi.org/10.1109/TSUSC.2018.2793284.

[27] Wen-Lin Chu, Chih-Jer Lin, Ke-Neng Chang, "Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine," Applied Sciences, vol. 21, pp. 45- 79, 2019.

[28] A. Boukhalfa, et al., "LSTM deep learning method for network intrusion detection system," International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, pp. 3315-3322, June 2020.

[29] https://www.kaggle.com/mrwellsdavid/unsw-nb15
[30] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," Neural Information Processing Systems, vol. 25, no 1. doi 10.1145/3065386.
[31] Igor Ševo, Aleksej Avramovic, "Convolutional Neural Network Based Automatic Object Detection on Aerial Images," IEEE Geoscience and Remote Sensing Letters, vol. 13(5), pp. 1-5, April 2016.
[32] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, Ingmar Posner. In 2017 IEEE International Conference on Robotics and Automation (ICRA). Singapore, pp. 1355-1361, 29 May-3 June 2017.
[33] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," 2016 Fourth International Conference on 3D Vision (3DV), pp. 565-571, 25-28 Oct. 2016.
[34] Pim Moeskops, Max A. Viergever, Adriënne M. Mendrik, Linda S. de Vries, Manon J.N.L. Benders, Ivana Išgum, "Automatic Segmentation of MR Brain Images With a Convolutional Neural Network," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1252-1261.
[35] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, Andrew D. Back, "Face Recognition: A Convolutional Neural-Network Approach," IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98-113, Jan. 1997.
[36] Yoon Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751, 25-29 October 2014.
[37] Nal Kalkbrenner, Edward Grefenstette, Phil Blunsom, "A Convolutional Neural Network for Modelling Sentences," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 655–665, 23-25 June 2014.
[38] Saad Albawi, Saad ALZAWI, Tareq Abed Mohammed, "Understanding of a Convolutional Neural Network," 2017 International Conference on Engineering and Technology (ICET), pp. 1-6, 21-23 Aug. 2017.
[39] Keiron O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks," arXiv, arXiv:1511.08458.
[40] Daniel Svozil, Vladimir Kvasnicka, Jiří Pospíchal, "Introduction to multi-layer feed-forward neural networks," Chemometrics and Intelligent Laboratory Systems, vol. 39(1), pp: 43-62, November 1997.
[41] Hassan Ramchoun, Mohammed Amine Janati Idrissi, Youssef Ghanou, Mohamed Ettaouil, "Multilayer Perceptron: Architecture Optimization and Training," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, no. 1, pp. 26-29, 2016.