2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)

LSTM-BA: DDoS Detection Approach Combining LSTM and Bayes

Yan Li
School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, China
liyan999Lee@163.com

Yifei Lu
School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, China
luyifei @njust.edu.cn

***Abstract*—The development of cyberspace brings both opportunities and threats, among which Distributed Denial of Service (DDoS) is one of the most destructive attacks. A mass of DDoS attack detection methods have been proposed. But more or less there are some problems, either the construction process is complex, or low accuracy, or poor generalization ability. To overcome these problems, in this paper, we propose a new DDoS detection method which combines the Long Short Term Memory (LSTM) and Bayes approach, referred to as LSTM-BA. Through LSTM method, we can identify parts of DDoS attacks with high confidence outputs from LSTM module. For those outputs with low confidence, we further use Bayes method for the second judgment to improve the accuracy. Our proposed method has been validated using publicly available datasets of ISCX2012. The results demonstrate that LSTM-BA has a better performance. More exactly, LSTM-BA achieves 98.15% detection accuracy, which is improved by 0.16% compared with the state-of-the-art method.**

Keywords—DDoS attack, DDoS detection, LSTM, Bayes

I. INTRODUCTION

Distributed Denial of Service (DDoS) is a highly destructive method of network attack. It has been developing and changing for more than a decade and has become a tool for different organizations and individuals to use in the network for extortion, revenge and even war. The economic and social costs of DDoS attacks are hard to estimate. Attackers make a large number of requests to the service provider and consume network resources for a long time, so that the requests from legitimate users cannot be processed at the service provider. Such attacks are reasonable requests on the face of it, therefore they will not be defended by system upgrades and patches as well as the intrusion detection system. Furthermore, it becomes more difficult to immediately distinguish between the abnormal requests and the normal requests when there are lots of normal requests. Therefore, it is important to detect DDoS attacks quickly and efficiently.

Researchers have proposed large number of detection mechanisms to address this problem. However, with the development of attack and defense technology, DDoS attacks are also emerging new features including more advanced and efficient technologies, more sustained processes, and more serious harm. This brings greater challenges to DDoS detection, protection, and mitigation [1]. The study of DDoS attack detection is confronted with the following difficulties: (1) attackers are adept in disguising themselves as real users, so that distinguish attack access from normal access becomes more difficult. (2) the diversification and customization of attack tools make it difficult to extract attack modes. (3) the rapid growth of attack traffic requires more efficient detection methods.

During the last decade, DDoS attack detection methods are mainly based on entropy, traffic flow features and machine learning. However, entropy methods and traffic pattern analysis methods both require extensive network knowledge and experiments to select appropriate statistical characteristics [2]. Moreover, due to poor generalization ability, new attacks cannot be detected effectively, and the model and threshold value need to be regularly updated. Machine learning approaches doesn't work well with large amounts of data, they still have drawbacks such as single detection attack type, omission and false alarm. In addition, due to the complexity of the algorithm, the existing DDoS attack detection methods cannot identify the network state timely. In order to effectively resist DDoS attacks, the DDoS defense mechanism should quickly detect and predict attacks while allowing legitimate users to access the targets [3].

Deep learning can solve these problems to some extent. Since traditional statistics and machine learning, even some deep learning methods fail to take historical patterns into account, the single-packet detection method cannot improve performance. The Long Short Term Memory (LSTM) is designed to avoid long dependency problems and can remember long historical information. By analyzing the flow information over a longer period of time, the accuracy of detection can be improved. Bayes method is also commonly used to detect DDoS attacks. It is simple and easy to implement, with high accuracy but poor real-time performance. Therefore, we use Bayes method to carry out secondary detection after deep learning, which can not only ensure the real-time performance of the system, but also improve the accuracy.

In this paper, we propose a model combining LSTM and Bayes method, referred to as LSTM-BA. LSTM-BA first learns the DDoS attack mode based on LSTM method using the network traffic. Through this module, we can gain a probability of prediction for DDoS attack. We can determine the DDoS attack with high prediction value and determine the normal traffic with low prediction value, whereas for those prediction value neither high nor low, we further use Bayes method to identify the DDoS. Experimental results show that our model improves the accuracy of 0.16% compared with the state-of-the-art method.

The rest of paper is organized as follows. Section II discusses the related work. In section III, we introduce the method of data preprocess and the construction of LSTM-BA. Section IV describes the experiment in detail and compares our results with other deep learning method and traditional machine learning method. Finally, we summarize our work in Section V.

This work was supported in part by National Natural Science Foundation of China under Grant No. 61702267 and Jiangsu Planned Projects for Post-doctoral Research Funds.

978-1-7281-5141-0/19/$31.00 ©2019 IEEE
DOI 10.1109/CBD.2019.00041

II. Related Work

To confront DDoS attacks, we eliminate the DDoS attacks through managements, detection, and alleviation. Management needs us to configure host, network and network equipment actively, eliminate DDoS security risks. Detection determines whether a DDoS attack has occurred through analyzing network traffic. Mitigation refers to reducing the impact of an attack when it has already occurred, thus ensuring the availability of the service.

In this section, we introduce some studies on DDoS detection and discuss the implications of our work.

There are some methods to detect attacks in specific scenarios: McDermott et al. use Bidirectional Long Short Term Memory to detect botnet activity in consumer Internet of Things (IoT) devices and networks [4]. Li et al. introduce a DDoS detection model based on deep learning in Software-Defined Network (SDN) environment [5]. Wang and Hu use quaternion feature vectors to distinguish attack traffic from normal traffic for switches in SDN data plane [6]. Han and Yang present a DDoS attack defense framework based on SDN data plane and control plane [7]. On the data plane, they use a monitoring algorithm to capture attack. On the control plane, they use a machine learning based DDoS attack classifier to locate a DDoS attack. Many existing detection mechanisms are based on Internet Protocol version four (IPv4) and do not apply to Internet Protocol version six (IPv6). In IPv6, DDoS attacks using Internet Control Message Protocol version six (ICMPv6) messages are the most common form. Elejla et al. introduce the DDoS attack based on icmpv6 in detail [8]. Cloud platforms are very popular these years, but its security is indeed a big problem. Because attackers can exploit vulnerabilities on the cloud platform, the detection of DDoS attacks is extremely difficult. Chung and Khatkar built a monitoring plane on a distributed programmable virtual switch to detect DDoS attacks on the cloud platform [9]. In addition to flood attack, shrew attack is a common form. They are periodic and low-rate. Thus, shrew attacks take a long time to detect. Yu and Kai propose a spectrum template matching method to combat shrew DDoS attacks [10].

In general scenarios, traditional methods such as: Xiang and Li use generalized entropy and information distance to measure the difference between normal traffic and attack traffic, so as to detect low-rate DDoS attacks [11]. Braga et al. propose a lightweight DDoS attack detection method based on traffic flow characteristics, they get feature information through the NOX controller [12]. In the cloud system, the reliability of the client is very important. Amuthan and Harikrishna use mean availability parameter to reduce the impact of DDoS on the dynamic client of the subnet, so as to ensure the security of cloud system [13]. Cheng et al. propose a method using abnormal network stream feature sequences in big data environment [3]. They set up a network traffic anomaly indicator set PDRA, and calculate the abnormal probability when consecutive PDRA values exceed the threshold. Feinstein et al. propose a method for calculating the entropy and frequency distributions of attributes [14]. Lee et al. analyze the characteristics of DDoS attacks, and select variables according to it, and then carry out clustering analysis to actively detect the attacks [15]. Chen et al. propose a distributed method based on change aggregation trees to detect traffic changes across multiple network domains [16]. In this way, we can detect it in the early stages of a DDoS attack, thereby minimizing the risk of attack. Umer et al. use stream-based intrusion detection [17]. They only analyze the packet header and do not analyze the packet payload, thus improving the detection speed.

However, traditional methods are slow in detection speed, require the selection of appropriate statistical features in advance, and have poor generalization abilities. Therefore, the machine learning-based methods emerge at the right moment: Mehmood et al. use Naïve Bayes classification algorithm for intrusion detection [18]. They deployed their system throughout the network for securing IoT against DDoS attacks. But machine learning methods are not suitable for large data volumes, so researchers propose some methods based on deep learning. Wang and Zhang propose a deep learning method to predict DDoS attacks by looking at the relevant streams of text in tweets [19]. They predict the likelihood of a DDoS attack by analyzing the content of the tweets, so that the attack can be blocked before it arrives. Yuan and Li use LSTM to detect attacks [20].

We also use deep learning method for detection. In addition, in order to improve the accuracy of detection, we use Bayes method to redetect part of the data.

III. LSTM-BA Model

In this section, we propose a method combining LSTM and Bayes approach, called LSTM-BA. The structure of LSTM-BA is shown in Figure 1. The original data is preprocessed and input into the LSTM module to identify DDoS attacks. For the data with poor reliability of LSTM prediction results, we use Bayes module to redetection. The outputs of the two modules are combined as LSTM-BA outputs.

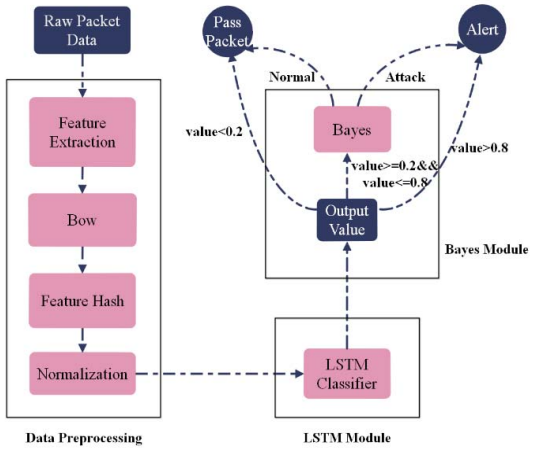


Fig. 1. Architecture of LSTM-BA example

A. Data Preprocess

We extract 10 features from the traffic, including source IP address, destination IP address, source port, destination port, protocol type, timestamp, duration, type of service, length and time to live. Because of the wide range of IP addresses, we use Bag of Word (BoW) [24] and feature hashing to convert IP address to a real vector. And for packets without port number

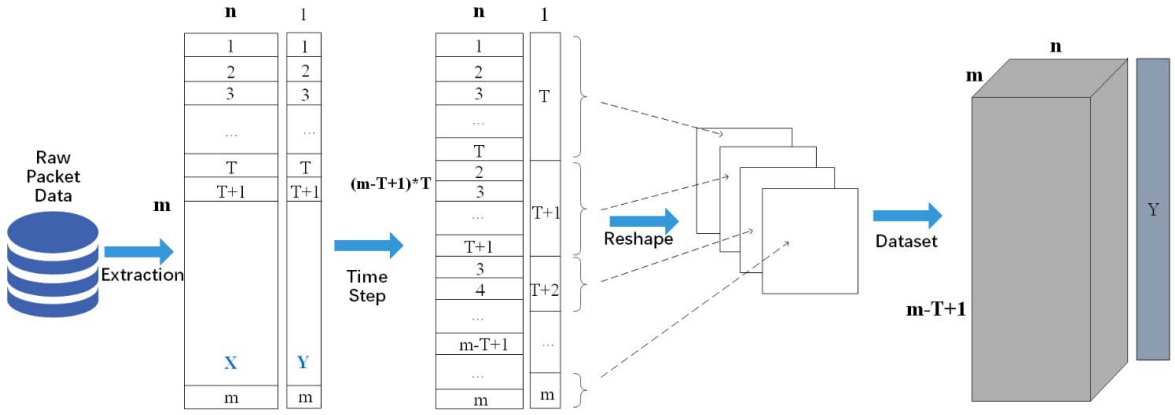


Fig. 2. Feature Extraction, Transformation and Reorganization

information, such as ICMP packets, we set their port number to 0.

After feature processing, we get a $m \times n$ data matrix, and a $m \times 1$ label matrix, where m indicates the number of packets and n indicates the number of transformed features. The label value of 0 represents normal traffic, the label value of 1 represents attack traffic. Since the LSTM module requires the input of a three-dimensional matrix (batch_size, time_step, input_dimension), we convert the two-dimensional matrix into a three-dimensional matrix $(m - T + 1) \times T \times n$. Where, T is the time window, representing the state of a packet associated with the previous $(T - 1)$ packets. Figure 2 illustrates the process of feature extraction, transformation, and reorganization. X is data matrix, Y is label matrix.

B. *LSTM Module*

We leverage the LSTM method to gain a prediction of DDoS. In this module, we enter the three-dimensional matrix into the input layer. After the operation of the hidden layer, the output layer outputs the prediction results.

LSTM takes the form of a repeating cell chain. The cell contains four types of interactive neural networks that interact in a special way to enable the network to remember historical information. LSTM protects and controls the state of cells through input gate, output gate and forget gate. Figure 3 depicts the architecture for LSTM. The right of the figure is a LSTM cell. The blue is forget gate, the yellow is input gate, the green is output gate, and the red means cell state renewal.

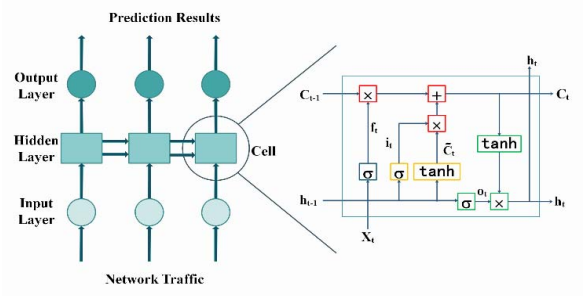


Fig. 3. Architecture of LSTM

The internal calculation formula of the LSTM cell is defined as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

where i, o, f mean input gate, output gate and forget gate, respectively. W is weight matrices and b is biases. $\tilde{C}$, C are the candidate state and the new state. h is output, x is input, t is input time t, σ denotes the Sigmoid function.

In the LSTM module, we use two hidden layers of 256 neurons, a full connection layer of 256 neurons, which activation function is ReLU, and a full connection layer of 1 neuron which activation function is Sigmoid. The values of all parameters are the optimal values that we have chosen after many comparative experiments.

The module uses the Sigmoid function to represent the prediction results of the last packet in the window:

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

A value smaller than 0.5 is considered normal traffic, and a value bigger than 0.5 is considered attack traffic. However, we found that the prediction value more closer to 0.5, the prediction accuracy more lower. So, for the data with poor reliability of the prediction result, we use Bayes method for the second discrimination to improve the accuracy. For other data, if it is determined to be attack traffic, we intercept it and send a warning to the service provider; otherwise, we forward it normally.

C. *Bayes Module*

In this module, we use Bayes method to calculate the probability of a packet being normal traffic and attack traffic.

Bayes method plays a significant role in predicting the probability of evidence accumulating things. And the LSTM module can get a probability value of whether the traffic is an attack or not. So it is convenient and appropriate to add a Bayes module after the LSTM module.

The detection process of Bayes module is as follows:

1) Let $x = \{a_1, a_2, \cdots, a_n\}$ be a set of data to be classified, and each a is a feature of x.

2) Class set $C = \{y_i\}$, $y = \begin{cases} 0, i = 1 \\ 1, i = 2 \end{cases}$ means normal traffic and attack traffic.

3) Calculate $P(y_1|x), P(y_2|x)$.

4) If $P(y|x) = max\{P(y_1|x), P(y_2|x)\}$, then $x \epsilon C_k$.

Now our task is calculate the conditional probabilities in step 3. Here's what we can do:

1) Find a set of packsts to be classified, that is, the set of data with poor prediction results reliability in LSTM module.

2) Since T is taken in our time window, calculate the conditional probability of each feature of $(T-1)$ packets in front of x under two categories.That is:

$$P(a_1|y_1), P(a_2|y_1), \cdots, P(a_n|y_1)$$
$$P(a_1|y_2), P(a_2|y_2), \cdots, P(a_n|y_2)$$

3) Since each feature is conditional independent, it can be deduced as follows according to bayes' theorem:

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

Because the denominator is constant for all categories, we can maximize the numerator. Since each feature is conditionally independent, it can be written as:

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)\cdots P(a_n|y_i)P(y_i)$$
$$= P(y_i)\prod_{j=1}^{n} P(a_j|y_i)$$

According to this theorem, we can calculate the probability of normal traffic and attack traffic in data x. The maximum probability determines the category of x. If the probabilities are the same, the category of the data is determined according to the judgment of the LSTM module. Similar to the LSTM module, if x is determined to be attack traffic, we intercept it and send a warning to the service provider; otherwise, we forward it normally.

IV. EXPERIMENTS

A. Data Process

We use intrusion detection evaluation dataset (ISCX2012) [23] training LSTM module. This dataset is a benchmark intrusion detection dataset includes seven days of network activity, as shown in TABLE I. We select the data of the fourth day, including 9,648,635 packets. It also provides label files. We set labels on the packets by comparing the fields of the packets in the .pcap file with the fields of the packets in the label file. Since most of them are normal traffic, we randomly select 120,000 normal packets and 120,000 attack packets to eliminate data skewness. The training set includes 108,000 normal packets and 108,000 attack packets, and the test set includes 12,000 normal packets and 12,000 attack packets. In each experiment we resample to eliminate errors.

TABLE I. INFORMATION OF ISCX2012 DATASET

Day	*Date*	*Description*	*Size (GB)*
Friday	11/6/2010	Normal Activity	16.1
Saturday	12/6/2010	Normal Activity	4.22
Sunday	13/6/2010	Infiltrating the network from inside + Normal Activity	3.95
Monday	14/6/2010	HTTP Denial of Service + Normal Activity	6.85
Tuesday	15/6/2010	Distributed Denial of Service using an IRC Botnet	23.4
Wednesday	16/6/2010	Normal Activity	17.6
Thursday	17/6/2010	Brute Force SSH + Normal Activity	12.3

B. Experimental Settings

In this section, we present the experimental results under different parameter settings. We train our model on the NVIDIA GTX 1050 GPU and repeat the experiment 10 times to reduce uncertainty in the dataset. We enter the training data into the LSTM module, and use 20% of the training data as a verification set to select the best model among training stage. The Sigmoid function is used to represent the prediction result of the last packet in the window. The predicted value smaller than 0.5 is discriminated as normal traffic, while the predicted value bigger than 0.5 is discriminated as attack traffic.

We test the accuracy of predicted values in the interval (left, right). Figure 4 shows the accuracy in different intervals. Obviously, the prediction accuracy becomes low when the predicted value is close to 0.5. Therefore, for these packets, the prediction results are unreliable. We need to make a second judgment on them.

The accuracy of Bayes method in detecting DDoS attack is about 92%. Considering the small amount of data may not reach such a high accuracy, we detect the data with lower accuracy. The prediction accuracy of the data in the range (0.2, 0.8) is 74%. After many experiments, we observed that the data in this interval are best detected by the secondary judgment. Therefore, for the data with predicted value between 0.2 and 0.8, we re-judge it for higher accuracy.

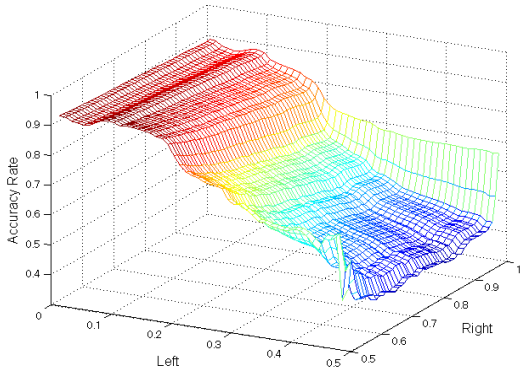


Fig. 4. Accuracy in different intervals

As we mentioned in Section III, when using LSTM module for prediction, we take time window T and consider the relationship between a packet and its previous $(T-1)$ packets. Therefore, in the second judgment, we also consider the relationship between a packet and its previous $(T-1)$ packets. Enter these T packets and make the judgment again according to Bayes method. In order to detect the accuracy with different time windows, we set T with the six different values: 5, 10, 20, 40, 70, and 100.

C. *Experimental Results*

We compare our model with Random Forest algorithm [24] and other deep learning method [20]. The experimental results were evaluated according to six indicators: Accuracy, Precision, Recall ,TNR (True Negative Rate), FPR (False Positive Rate) and F1 Score. They are defined as follows:

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

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

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

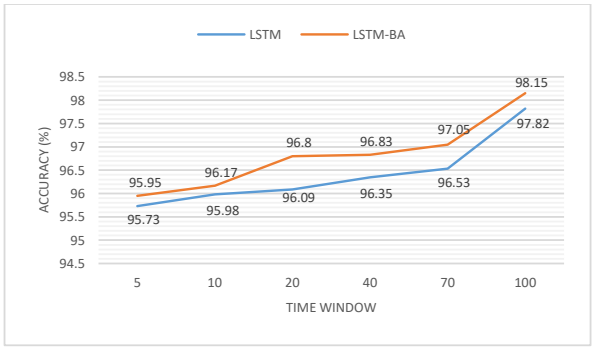
$$\text{TNR} = \frac{TN}{TN + FP}$$

$$\text{FPR} = \frac{FP}{TN + FP}$$

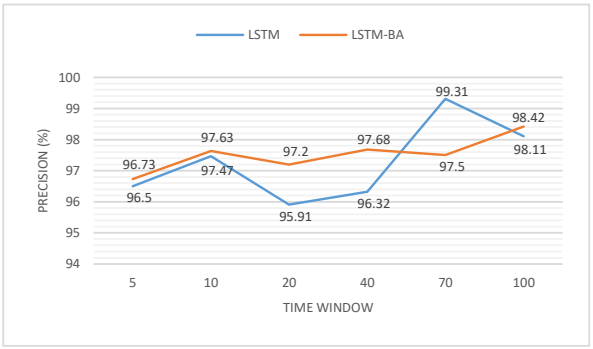
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP (True Positives) refer to attack packets correctly classified as attack, TN (True Negatives) refer to normal packets correctly classified as normal, FP (False Positives) refer to normal packets incorrectly classified as attack, FN (False Negatives) refer to attack packets incorrectly classified as normal.

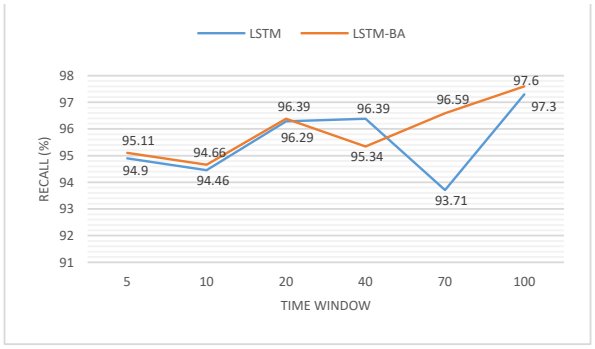
Figure 5 illustrates the results of LSTM module without Bayes module and LSTM-BA. We find LSTM-BA has a better performance, and accuracy increases when time window increases. But detection time increases with time window too, so that loss of detection time and computational complexity can offset the benefits of performance improvement. As we know, no one set the value of time window over 100 [20].



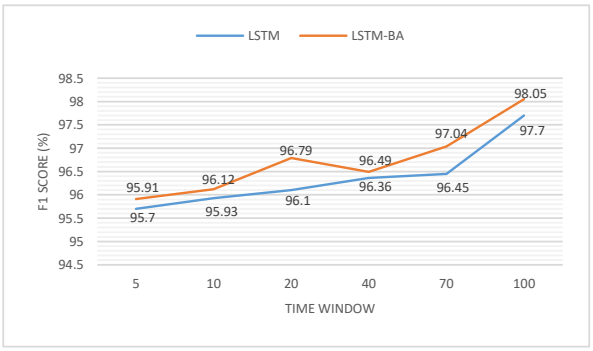
(a) Accuracy



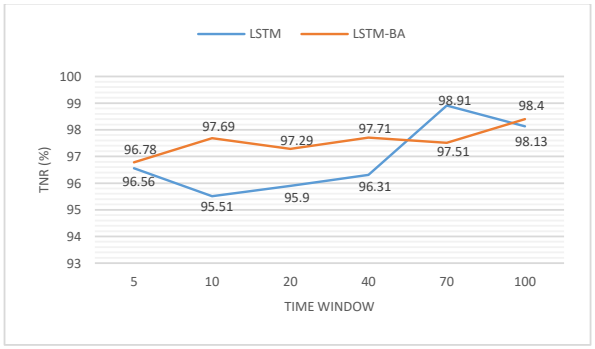
(b) Precision



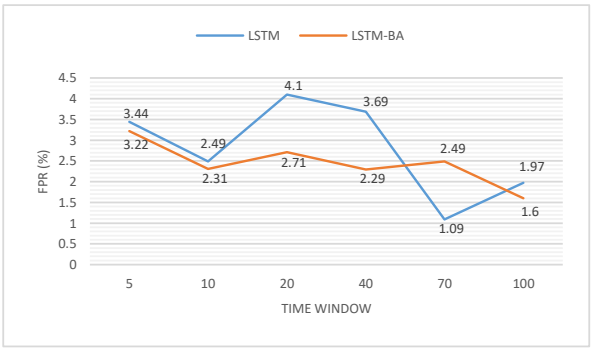
(c) Recall



(d) F1 Score



(e) TNR



(f) FPR

Fig. 5. Results of LSTM module and LSTM-BA with different time window

Figure 5 illustrates that precision and recall are not always rise. Even the precision and recall of LSTM-BA is not always higher than LSTM. However, F1 score, the harmonic mean of the precision and recall, shows an upward trend. And the F1 score of LSTM-BA is always higher than LSTM. That means the performance of the model will increase with the increase of the time window. In addition, the Bayes module can improve the performance. When time window is 100, our best model achieves 98.15% accuracy, 98.42% precision, 97.6% recall, 98.05% F1 score, 98.4% TNR and 1.6% FPR.

Then, we compare our model with other methods including DeepDefense [20] and Random Forest which are implemented by ourselves. TABLE II shows the result. We observe that LSTM-BA has the highest accuracy and F1 score. In other words, our model works better than other methods when detecting DDoS attacks.

TABLE II. PERFORMANCE COMPARISON AMONG DEEPDEFENSE AND LSTM-BA

ModelName	*Accuracy*	*Precision*	*Recall*	*F1 Score*
LSTM-BA	98.15%	98.42%	97.6%	98.05%
DeepDefense	97.99%	98.1%	97.88%	97.99%
Random Forest	93.12%	97.51%	88.52%	92.8%

Finally, to verify the generalization of LSTM-BA, we test its performance on data of the fifth day of the ISCX2012 dataset. The results are shown in TABLE III.

TABLE III. PERFORMANCE COMPARISON AMONG THE FOURTH DAY AND THE FIFTH DAY

Date	*Accuracy*	*Precision*	*Recall*	*F1 Score*
14/6/2010	98.15%	98.42%	97.6%	98.05%
15/6/2010	98.03%	98.21%	97.55%	97.87%

We find although performance indicators have declined in the new data, the decline is small and the results are still good. This proves LSTM-BA is generalized.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a DDoS attack detection model combining LSTM and Bayes method. Experimental results show that LSTM-BA can improve the performance of identifying DDoS attacks and normal traffic. LSTM-BA increases accuracy by 0.16% compared with DeepDefense. After training with the dataset, in the real network environment, we can directly enter the network traffic into the LSTM-BA model for DDoS detection. Since DDoS attacks are constantly coming in new forms and tools, and LSTM-BA can use the detected data to train its model to ensure that the model is continuously updated. Therefore, LSTM-BA performs better in the face of new attacks than other methods.

In future work, we plan to increase the type of DDoS attack and diversity of model settings to test the robustness of LSTM-BA in different environments.

REFERENCES

[1] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39-53.

[2] Zhou, C. V., Leckie, C., & Karunasekera, S. (2010). A survey of coordinated attacks and collaborative intrusion detection. Computers & Security, 29(1), 124-140.

[3] Cheng, R., Xu, R., Tang, X., Sheng, V. S., & Cai, C. (2018). An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. Computers, Materials & Continua, 55(1), 095-095.

[4] McDermott, C. D., Majdani, F., & Petrovski, A. V. (2018, July). Botnet detection in the internet of things using deep learning approaches. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[5] Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., & Gong, L. (2018). Detection and defense of DDoS attack–based on deep learning in OpenFlow‐based SDN. International Journal of Communication Systems, 31(5), e3497.

[6] Wang, Y., Hu, T., Tang, G., Xie, J., & Lu, J. (2019). SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking. IEEE Access.

[7] Han, B., Yang, X., Sun, Z., Huang, J., & Su, J. (2018). OverWatch: a cross-plane ddos attack defense framework with collaborative intelligence in SDN. Security and Communication Networks, 2018.

[8] Elejla, O. E., Belaton, B., Anbar, M., & Alnajjar, A. (2018). Intrusion detection systems of ICMPv6-based DDoS attacks. Neural Computing and Applications, 30(1), 45-56. Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013).

[9] Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE transactions on dependable and secure computing, 10(4), 198-211.

[10] Chen, Y., & Hwang, K. (2006). Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. Journal of Parallel and Distributed Computing, 66(9), 1137-1151.

[11] Xiang, Y., Li, K., & Zhou, W. (2011). Low-rate DDoS attacks detection and traceback by using new information metrics.IEEE transactions on information forensics and security,6(2), 426-437.

[12] Braga, R., de Souza Mota, E., & Passito, A. (2010, October). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In LCN (Vol. 10, pp. 408-415).

[13] Amuthan, A., & Harikrishna, P. (2019). Mean Availability Parameter-Based DDoS Detection Mechanism for Cloud Computing Environments. In Wireless Communication Networks and Internet of Things (pp. 115-122). Springer, Singapore.

[14] Feinstein, L., Schnackenberg, D., Balupari, R., & Kindred, D. (2003, April). Statistical approaches to DDoS attack detection and response. In Proceedings DARPA information survivability conference and exposition (Vol. 1, pp. 303-314). IEEE.

[15] Lee, K., Kim, J., Kwon, K. H., Han, Y., & Kim, S. (2008). DDoS attack detection method using cluster analysis. Expert systems with applications, 34(3), 1659-1665.

[16] Chen, Y., Hwang, K., & Ku, W. S. (2007). Collaborative detection of DDoS attacks over multiple network domains. IEEE Transactions on Parallel and Distributed Systems, 18(12), 1649-1662.

[17] Umer, M. F., Sher, M., & Bi, Y. (2017). Flow-based intrusion detection: Techniques and challenges. Computers & Security, 70, 238-254.

[18] Mehmood, A., Mukherjee, M., Ahmed, S. H., Song, H., & Malik, K. M. (2018). NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. The Journal of Supercomputing, 74(10), 5156-5170.

[19] Wang, Z., & Zhang, Y. (2017, August). DDoS Event Forecasting using Twitter Data. In IJCAI (pp. 4151-4157).

[20] Yuan, X., Li, C., & Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 1-8). IEEE.

[21] Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

[22] Weinberger, K., Dasgupta, A., Attenberg, J., Langford, J., & Smola, A. (2009). Feature hashing for large scale multitask learning. arXiv preprint arXiv:0902.2206.

[23] "Unb iscx intrusion detection evaluation dataset," http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html.

[24] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-3