DeepDefense: Identifying DDoS Attack via Deep Learning

Xiaoyong Yuan*, Chuanhuang Li†, Xiaolin Li*
*Large-scale Intelligent Systems Laboratory, University of Florida
†Zhejiang Gongshang University

Abstract

Distributed Denial of Service (DDoS) attacks grow rapidly and become one of the fatal threats to the Internet. Automatically detecting DDoS attack packets is one of the main defense mechanisms. Conventional solutions monitor network traffic and identify attack activities from legitimate network traffic based on statistical divergence. Machine learning is another method to improve identifying performance based on statistical features. However, conventional machine learning techniques are limited by the shallow representation models. In this paper, we propose a deep learning based DDoS attack detection approach (called *DeepDefense*). Deep learning approach can automatically extract high-level features from low-level ones and gain powerful representation and inference. We design a recurrent deep neural network to learn patterns from sequences of network traffic and trace network attack activities. The experimental results demonstrate better performance of our model compared with conventional machine learning models. We reduce the error rate from 7.517% to 2.103% compared with conventional machine learning method in the larger data set.

KEYWORDS: DDoS attack; deep learning; recurrent neural network; LSTM

I. Introduction

A Denial of Service (DoS) attack denies the access of other legitimate users to shared services or resources [1]. Distributed Denial of Service (DDoS): attackers usually use multiple distributed computer resources to launch a coordinated DoS attack against one or more targets [2]. They mainly aim to system resources and network bandwidth, ranging from Network layer to Application layer. Since the first DDoS attack occurred in 1999 [3], DDoS has become a critical, widespread, and rapidly evolving threat in the world. According to a survey from Radware, DDoS is currently the largest threat (50% respondents in the survey) for organizations [4]. There are 24 DDoS attack vectors witnessed by Akamai in Q4 2015. Compared with Q4 2014, total DDoS attacks increase by 148.85%, and multi-vector attack largely increased [5]. Currently, main attack vectors include UDP flood, HTTP flood, SYN flood, ICMP, DNS, etc and pose serious threats to both systems and networks [6].

DDoS detection is one of the main DDoS defense mechanisms. However, it is hard to detect DDoS attack automatically because in most cases attack traffic is very similar to legitimate traffic and attackers try to mimic flash crowds. An attack activity with insufficient traffic can even be seen as a legitimate one in early stages [7]. Many researchers try statistical machine learning methods to identify DDoS attacks (discussed in Section II). Machine learning methods identify DDoS attacks based on statistical features and perform better than statistical ways. However, they are prone to several drawbacks: 1) requiring extensive network expertise and experiments in DDoS to select proper statistical features; 2) limited to only one or several DDoS attack vectors; 3) requiring updating its model and threshold value to satisfy the changes of systems and attack vectors; 4) vulnerable to slow attack rate [8].

In this paper, we propose a deep learning based approach called *DeepDefense* to detect DDoS attacks from legitimate network traffic at victim end and alleviate the aforementioned issues. We train our deep learning models with a large-scale dataset, *UNB ISCX Intrusion Detection Evaluation 2012 DataSet* (We use ISCX2012 for short in rest paper) [9] [10], to solve complicated recognition problems. In the experiments, we process two days' network traffic from ISCX2012 to train both shallow machine learning models and our deep learning models. *DeepDefense* leverages different neural network models: Convolutional Neural Network (CNN) [11], Recurrent Neural Network (RNN), Long Short-Term Memory Neural Network (LSTM) [12], and Gated Recurrent Unit Neural Network (GRU) [13]. These methods are proved to greatly improve the performance in many domains when training large data sets. Deep learning approach is a natural fit for large scale of network traffic. LSTM and GRU can also help us gain context of network packets, especially the long and short term patterns in DDoS attack sequences. From the experimental results, our best deep learning model reduces the error rate by 39.69% compared with shallow machine learning methods in a small dataset. In a large dataset, we can even reduce error rate from 7.517% to 2.103%. This shows its capability of learning from historical network packets. Moreover, *DeepDefense* also outperforms shallow machine learning methods in terms of generalization.

The rest of paper is organized as follows. Section II discusses the related work. We propose our deep learning based model, data preprocessing methods, and overall architecture in Section III. Section IV elaborates the experiments on the ISCX2012 data set and compare our results with shallow machine learning method. Finally, Section V concludes our work.

978-1-5090-6517-2/17/$31.00 ©2017 IEEE

II. Related Work

A. DDoS Attacks and Countermeasures

The design of open, scalable and independent security on the Internet exploits the vulnerability to DDoS attack [8]. Host's resources and network bandwidth are two main targets of DDoS attacks. Most attacks aim at the defect of protocols and applications: SYN flood, UDP flood, ICMP flood, SIP flood, etc. Some attacks like UDP flood, ICMP flood deplete the network bandwidth. Others like SYN flood, SIP flood exhaust a victim's system resource (e.g., CPU, memory) as well [14]. For example, a victim doesn't need to handshake when receiving packets via UDP. In a UDP flood attack, an attacker sends the packets to some random or specified ports to attack these ports and saturating the network resources. DDoS attacks also take advantage of techniques like IP spoofing, network amplifier/reflector, the combination of attack methods to avoid detection and prompt their influence [7]. DNS amplification is an example of such techniques. An attacker can fill up the victim's bandwidth that is ten times greater than its bandwidth [15].

To defense DDoS attacks, some solutions leverage both preventive and reactive mechanisms to alleviate the impact of DDoS attack in victim network, intermediate network and source network [8]. Mostly used mechanisms include attack prevention, attack detection, and attack reaction. Attack prevention tries to filter ingress and egress traffic before the attack causes damage. Attack reaction aims to minimize the loss of DDoS attack. Existing approaches for DDoS attack detection include statistical methods and machine learning methods. We introduce the statistical methods in this subsection and overview machine learning approaches in the next subsection.

D-WARD, a DDoS defense system, is deployed at source-end networks and monitors two-way traffic. D-WARD presents the ability to detect DDoS attack based on a statistical comparison between legitimate flow and DDoS flow of different protocols [16]. Bhuyan and Kalita compare four important information entropy measures (i.e., Hartley entropy, Shannon entropy, Renyin++s entropy and Renyin++s generalized entropy) in DDoS detection [17]. According to these four information entropy methods, they calculate the information distance in the network traffic and detect both low-rate and high-rate DDoS attack. Chen identifies DDoS attack by two statistical t-testing of the SYN arrival rate and the number of SYN and ACK packets [18]. However, most statistical methods perform well in specific DDoS attack methods and need work on feature selection and preprocessing metrics.

B. DDoS Defense Based on Machine Learning

Many machine learning algorithms are applied to DDoS defense especially in the phase of anomaly detection. Most frequently used algorithms include Naïve Bayes, Neural Network, Support Vector Machine, Decision Tree, K-Nearest Neighbor.

Figure 1 depicts a general architecture for DDoS attack detection system. In the first step, network traffic is filtered by filtering rules and stored in the database. Features will be extracted from the traffic (e.g., packet rate, protocol type, etc.) in the next step. These features are normalized to speed up the training process. Training data will be trained by some machine learning algorithms. Each packet in real network traffic will be classified as a DDoS attack or a legitimate one. In the last step, the system drops the DDoS packets detected by machine learning algorithm and updates its filtering rules.

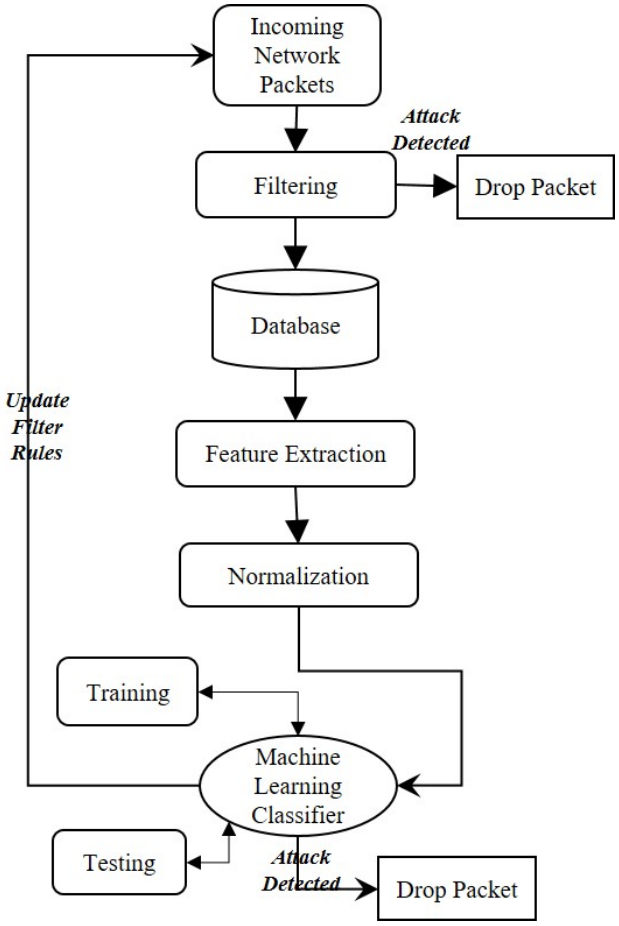


Figure 1: Architecture of DDoS Attack Detection Based on Machine Learning.

Xu, Sun and Huang use Hidden Markov Models (HMM) and reinforcement learning to differentiate legitimate traffic from DDoS attack based on monitored source IP addresses [19]. Detection agents are placed in the mediate network nodes or near the sources of DDoS attacks. HMM is proposed to compute the probability of a particular observation sequence of new IP addresses. They claim that most source IP addresses are new to the victim during a DDoS attack. Similar to their work, Berral et al. collect traffic information from the intermediate network and let each node learn independently and share information [20]. Naïve Bayes algorithm is equipped in each node and detects DDoS attack based on source address, destination address, and shared information. Their approach can reduce the reaction time and degrade the impact of DDoS attack. Shon et al. propose a DDoS detection method by using

both Genetic Algorithm (GA) and Support Vector Machine (SVM) [21]. They include more fields of network traffic by feature selection based on GA. Then they classify the packets by SVM to detect DDoS attack.

Many researchers (e.g., Manikopoulos and Papavassiliou, Seufert and OBrien, and Kumar et al.) leverage neural network to detect potential DDoS attack based on information in each network packet [22] [23] [24]. Manikopoulos and Papavassiliou combine the statistical test and neural network. They first conduct the Klomogrov-Smirnov test to get similarity measures and then classify the network packets using five different types of neural networks. They find back propagation and perceptron-back propagation hybrid approach have a stronger classification. Seufert and O'Brien detect unknown (Zero-Day) attacks by a set of probes extracting features from different protocol layers and system resources [23]. They claim that focusing on both network traffic and system resources can reduce the chance of false positive (dropping legitimate requests) because the system becomes overloaded only due to DDoS attack. Kumar et al. focus on the false alarm of DDoS attack detection [24]. To minimize the cost of misclassification, multiple Resilient Back Propagation models are proposed get initial results and then they select the best one among RBP models using Weighted Majority Voting and Weighted Product Rule based on Q-statistics, a metric measuring misclassification, for these results.

III. Deep Learning based Approach

It is hard to detect low rate attack because it looks similar to the legitimate network traffic from the victim-end. Meanwhile, DDoS attacks toward victim systems must be generated over time. Otherwise, it won't be malicious to the network/system resources. This suggests the importance of historical information in DDoS detection. Single-packet based detection method can't improve the performance due to the missing historical pattern in the learning model.

Therefore, we design the *DeepDefense* approach to identify DDoS attack based on Recurrent Neural Network (RNN) (e.g., LSTM, GRU). RNN has achieved dramatic improvement in speech recognition [25], language translation [26], speech synthesis [27], and other sequential data [28] [29].

Our detection approach utilizes a sequence of continuous network packets and is able to learn the subtle difference between attack traffics and legitimate ones. Historical information is fed into RNN model to identify DDoS attack. It helps to find repeated patterns representing DDoS attacks and locate them in a long term traffic sequence.

Another advantage of RNN is the independence from input window size. Window size used in previous machine learning methods usually is task-dependent. This brings a limitation for these methods to detect different types of attacks. What's more, it becomes hard to train a long-term sequence for conventional machine learning methods. However, RNN (especially gated RNN, i.e., LSTM, GRU) has shown the capability to solve these problems [30].

We present four RNN models in this section. From our experimental results, all of them outperform shallow machine learning models. RNN demonstrates the effectiveness of identifying attacking network packets. The performance improves with the increase of the length of history. We also adopt CNN in our model to gain the local correlations among network fields.

A. Feature Extraction and Transformation

We first extract 20 network traffic fields from the ISCX2012 data set. Table I shows examples and types of these fields. We use them as the training features. Different from other machine learning methods applied on DDoS detection, we do not need to select statistical features in our model.

Table I: Examples and Types of 20 network traffic fields

Field	Field Example	Field Type
frame.encap_type	1	Boolean
frame.len	805	Numerical
frame.protocols	eth:ip:tcp:http:data:data:data:data-text-lines	Text
http.time	0	Numerical
icmp.length	203	Numerical
icmp.type	3	Boolean
irc.request.command	PONG	Text
irc.response.command	462,JOIN,353,366	Text
tcp.ack	2.692e+03	Numerical
tcp.analysis.ack_rtt	0	Numerical
tcp.analysis.bytes_in_flight	1.460e+03	Numerical
tcp.analysis.duplicate_ack_num	1	Numerical
tcp.dstport	2090	Boolean
tcp.flags.urg	0	Boolean
tcp.len	751	Numerical
tcp.srcport	80	Boolean
tcp.window_size	12864	Numerical
udp.dstport	47666	Boolean
udp.length	97	Numerical
udp.srcport	47521	Boolean

We classify three types of fields: text fields, boolean fields, and numerical fields. We transform the representation for the content of most text and boolean fields. For boolean fields like TCP/UDP port number, we transform them to binary. For example, we encode TCP and UDP port number to a 16-bit binary list. For text fields, we transform them via Bag of Word (BoW) method [31]. To deal with a large dataset and conquer the memory scalability, we apply hashing method in the conversion of BoW [32].

After feature transformation, we get a $m \times n'$ matrix, where m indicates the number of packets and n' indicates the number of new features after transformation. In order to learn patterns in both long and short term, we use a sliding window to separate continuous packets and reshape the data into a series of time windows with window size T. The label y in each window illustrates the last packet. After reshaping, we have a

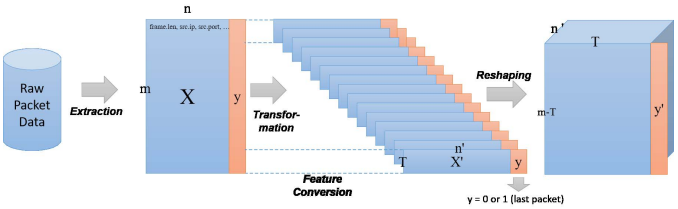


Figure 2: Feature Extraction and Transformation

three-dimensional matrix with shape $(m-T)\times T\times n'$. Figure 2 illustrates the workflow of feature extraction, transformation, and reshaping.

In this way, we change the features from conventional packet-based to window-based, by which we can learn network patterns from both previous $(T-1)$ packets and current packet.

B. Bidirectional Recurrent Neural Network Architecture

In this section, we design a Bidirectional Recurrent Neural Network in the model. Each direction includes two sequence-to-sequence Recurrent Neural Layers. Recurrent Neural Layers help us trace the history from previous network packets. To eliminate scaling issues in deep RNN network, we try both LSTM and GRU in this paper. In specific, LSTM aims to overcome vanishing gradient problem of RNN and uses a memory cell to present the previous timestamp [12]. GRU is a simpler variant of the standard LSTM and can be trained faster because of fewer parameters. Current modified LSTM usually includes three gates in each cell: input, forget, and output. They are calculated as follows:

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

where x_t is the input at time t, W_i, W_C, W_f, W_b are weight matrices, b_i, b_C, b_f, b_o are biases, $C_t, \tilde{C}_t$ are the new state and candidate state of memory cell, f_t, o_t are forget gate and output gate. We design 64 neurons in each cell and model the neuron's output f as a nonlinear activation function:

$$f(x) = \tanh(x)\,. \tag{1}$$

We concatenate output from layers in both directions and connect it to latter layer. Fully Connected Layers are designed followed by Recurrent Neural Network. We set the output function as Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}}\,. \tag{2}$$

The deep learning model use Sigmoid function to indicate the prediction of the last packet in the whole sequence. To capture the local information and simplify the deep neuron network, we try to stack 1-Dimensional Convolutional Neural Layers before Recurrent Neural layers with Rectified Linear Unit (RELU) as activation function [33]. Kernel size of Convolutional Neural Layers is 3 with stride 1. Every two RNN layers and every fully connected layer are followed by a Batch Normalization Layer to accelerate deep neuron network training [34]. Figure 3 summarizes the overall network architecture for *DeepDefense*. Detailed specification of the neural network model is explained in Section IV.

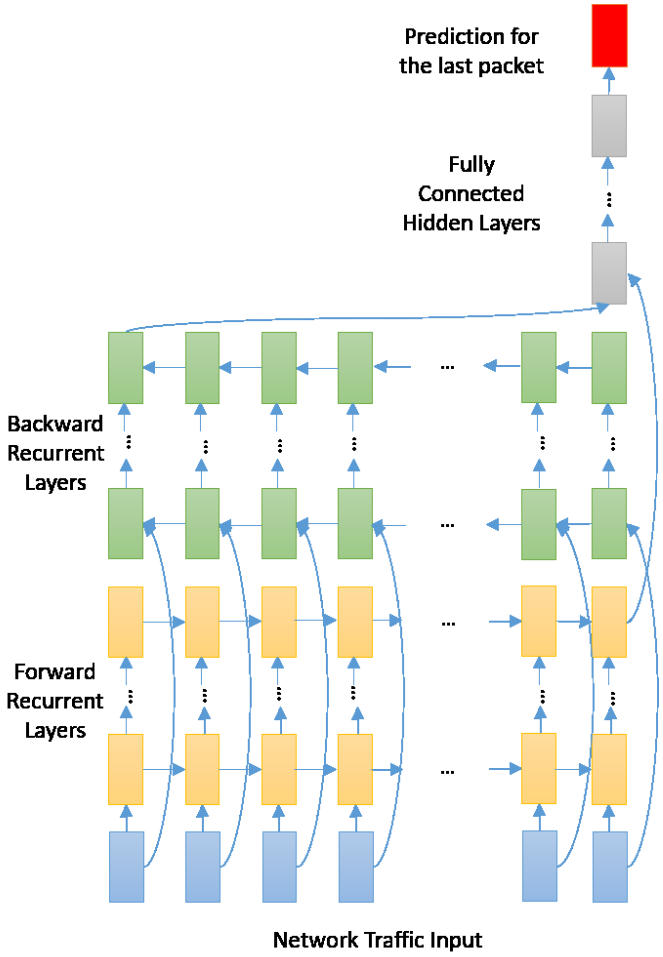


Figure 3: Overall Network Architecture for *DeepDefense*

In our experiments, we tried 1, 5, 10, 50, and 100 timestamps (window size) in the memory. A Larger number of window size can store longer attack time sequences and reflect more integrated attack activities. To our best knowledge, no one has traced back more than 100 packets for DDoS detection.

IV. Experiments

A. Dataset Description and Preprocessing

We evaluate *DeepDefense* on ISCX2012 dataset. ISCX2012 recorded 7 days' network traffic (legitimate and malicious). During these 7 days, DDoS attack occurred in 6/14/2010 and 6/15/2010. We extract these two days' network traffic and name them *Data14* and *Data15*. *Data14* includes

9,648,653 packets (6.85GB). *Data15* includes 34,983,043 packets (23.4GB).

The ISCX2012 dataset also contains a list of labels. This list shows legitimate or attack network activities with its corresponding information (e.g., name, time period, source IP, destination IP, source port, destination port, etc.). Major attack vectors are listed in Table II. Because each attack is only listed in the description, we first match the attack list with network traffic and label each attack and legitimate packet.

Table II: Attack Description in *Data14* and *Data15*

Data14		*Data15*	
Attack Name	Frequency	Attack Name	Frequency
HTTPWeb	2165	HTTPWeb	37158
Unknown_TCP	896	IRC	212
SecureWeb	92	Unknown_TCP	5
MiscApplication	77	DNS	1
ICMP	55	SecureWeb	1
Flowgen	50	SMTP	1
IMAP	32	/	/

Because most network traffics in *Data14* and *Data15* are legitimate ones, to eliminate data skewness, we collect all attack packets and randomly choose legitimate ones to match the number of attack packets. Total number of training samples of *Data14* and *Data15* are 15,176 and 233,450. We conduct this re-sampling work before every experiment.

B. Experimental Settings and Results

In this section, we present the experimental results derived from different deep neural network models and conventional machine learning model with various window sizes.

We train our models on NVIDIA Tesla M40 GPUs with 12GB memory and repeat the experiments ten times to decrease uncertainty from data sets. We separate data into training and testing sets with ratio 9:1. We feed the training data into our deep network models and use 10% of training data as a validation set to select the best model among training epochs. As discussed in Section III, Table III describes the specifications of models used in experiments. To simplify the expression, we will name each model with the "ModelName" listed in Table III to explain our experimental results.

We compare among different RNN models to select the best model by evaluating the results with same features and data. The comparison is also conducted between RNN model and Random Forest (RF) algorithm [35]. Random Forest is a widely used shallow machine learning method in practical. We choose Random Forest as the baseline of conventional machine learning models in the experiments. We evaluate the experimental results according to six metrics: error rate, accuracy, precision, recall, F1 score[1], and AUC[2].

1) Comparison among Recurrent Neural Network models: We first compare the performance of four different Recurrent Neural Network models. In this experiment, we set the timestamp of RNN's inputs as 100, as it achieves the best accuracy, according to the analysis in Section IV-B2. Table IV shows the results of comparison. We highlight the best results in different metrics and datasets.

From the experimental results, we find all these four Recurrent Neural Network models achieve almost the same performance. LSTM model shows slightly higher accuracy (97.996%), recall(97.887%) score and less error rate (2.004%) in *Data14*. 3LSTM is the best model for *Data15*. However, LSTM and GRU are close to 3LSTM less than 1%. Because deeper neural network model can extract higher-level feature and illustrates the better representation of large and complex data, 3LSTM outperforms LSTM and GRU in *Data15*, which is a larger dataset compared with *Data14*.

The experimental results show that LSTM and GRU both share the superiority of the gated units and outperform the shallow machine learning methods. There doesn't exist much difference between these two models. This meets the previous analysis of LSTM and GRU research [13] [36].

Because there are no spatial features in the problem, Convolutional Neural Network doesn't show its capacity of encoding spatial information and local connectivity. Stacking CNN in the first few layers can't gather useful features and exceed normal LSTM and GRU model in the experiments.

Figure 4a and 4b illustrate the training iterations of the 3LSTM model on two datasets in an experiment. From the small gap of error rate between the training set and validation set, it shows not much over-fitting in the training iterations and performance of 3LSTM model can improve quickly after only a few iterations of training.

2) Comparison among Recurrent Neural Network and Random Forest: We first compare the performance of Recurrent Neural Network model and Random Forest model in this section. Then, we analyze the impact of window size on these two models. In the end of this section, we apply these two models on two datasets to verify their generalization.

According to the analysis in Section IV-B1, we choose LSTM model as a baseline of various Recurrent Neural Network models and compare LSTM with Random Forest. Figure 5 and 6 illustrate the performance comparison between LSTM model and Random Forest model with different window sizes.

On *Data14*, our best LSTM model (with window size of 100) achieves 2.004% error rate, 97.997% F1 score, and 99.287% AUC. In comparison, the best Random Forest model (with window size of 1) achieves 2.800% error rate, 97.117% F1 score, and 97.18% AUC. Our *DeepDefense* approach reduces the error rate by 39.69% on *Data14*. On the larger data set *Data15*, the gap reaches in a higher level. LSTM achieves 2.103% error rate, 97.891% F1 score, and 99.226% AUC. However, Random Forest only achieves 7.517% error rate, 92.518% F1 score, and 92.483% AUC. Our *DeepDefense* approach reduces the error rate from 7.517% to 2.103%.

We observe that with the increase of window size, the performance of LSTM model improves on two datasets. LSTM

[1]Wikipedia, F1 score, https://en.wikipedia.org/wiki/F1_score

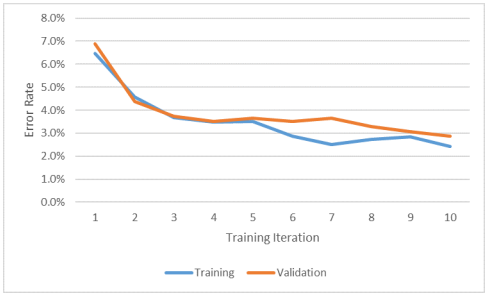
[2]Wikipedia, AUC, Area under the curve, https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve

Table III: Specifications of Different RNN models

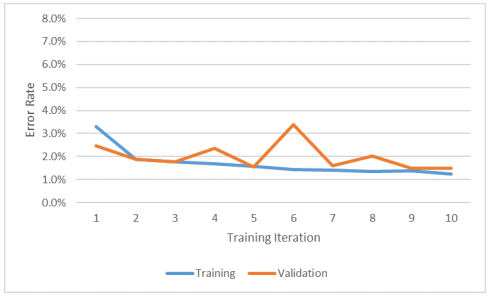
ModelName	LSTM	CNNLSTM	GRU	3LSTM
# LSTM/GRU layer	4	4	4	6
# neuron	64	64	64	64
activation function	tanh	tanh	tanh	tanh
# CNN Layer	/	2	/	/
# neuron	/	128	/	/
activation function	/	relu	/	/
# Fully Connected Layer	2	2	2	2
# neuron	128, 1	128, 1	128, 1	128, 1
activation function	relu, sigmoid	relu, sigmoid	relu, sigmoid	relu, sigmoid

Table IV: Performance Comparison among Different RNN models

	ModelName	Error Rate	Accuracy	Precision	Recall	F1	AUC
Data14	LSTM	2.004%	97.996%	98.108%	97.887%	97.997%	99.287%
	CNNLSTM	4.104%	95.896%	97.534%	94.208%	95.831%	98.465%
	GRU	3.208%	96.792%	98.417%	95.018%	96.681%	99.044%
	3LSTM	3.505%	96.495%	97.688%	95.232%	96.437%	98.805%
Data15	LSTM	2.103%	97.897%	97.941%	97.844%	97.891%	99.227%
	CNNLSTM	3.561%	96.439%	96.276%	96.888%	96.482%	99.199%
	GRU	1.917%	98.083%	98.118%	98.054%	98.085%	99.299%
	3LSTM	1.590%	98.410%	98.342%	98.476%	98.408%	99.450%



(a) *Data14*



(b) *Data15*

Figure 4: Error Rate for Training iterations using 3LSTM

with window size of 100 achieves the lowest error rate. This shows the capability of Recurrent Neural Network in long-term sequences. Recurrent Neural Network is proved to learn patterns from long-term network traffic. However, the performance of Random Forest model drops quickly, especially in a big dataset. Random Forest is good at single-packet based prediction than the multi-packet one. Historical features cannot help Random Forest to identify the patterns in the sequence.

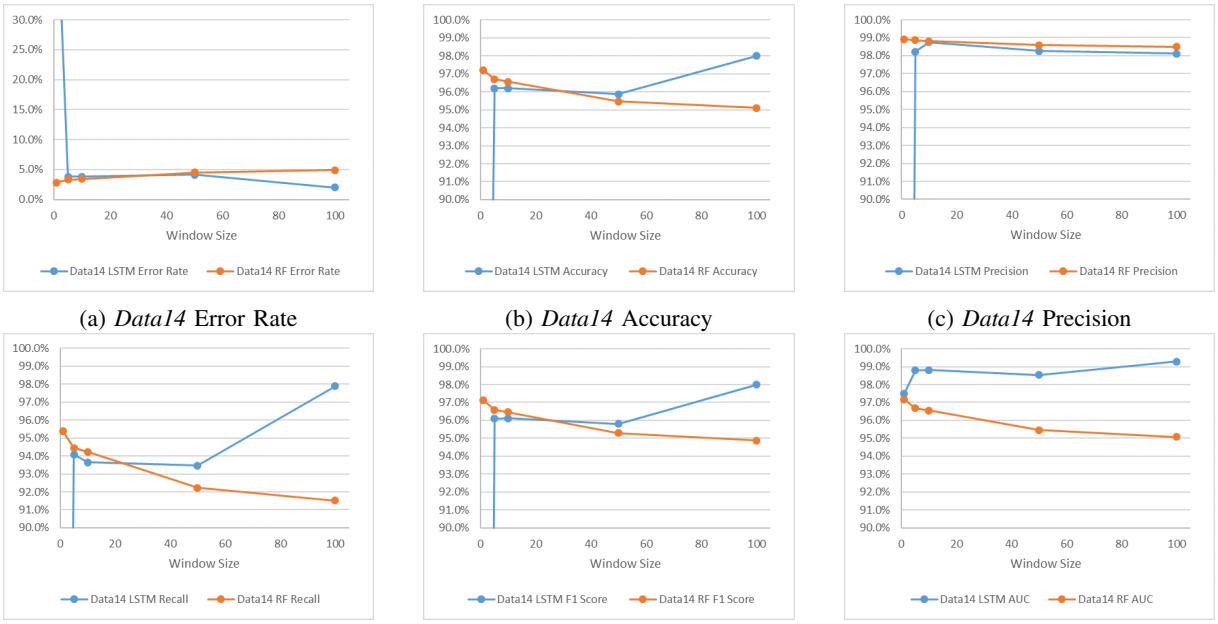
To verify the generalization of LSTM and Random Forest method, we follow the best method mentioned above (LSTM with window size 100 and Random Forest with window size 1) and train these models on both two datasets. Table V shows that the performance of LSTM model doesn't decrease during generalization. However, Random Forest can't adapt to two different datasets very well.

V. Conclusion and Future Work

In this paper, we proposed a deep learning based DDoS detection approach, *DeepDefense*. It helps improve the performance of identifying DDoS attack traffic. We formulate the DDoS detection as a sequence classification problem and transform the packet-based DDoS detection to the window-based detection. *DeepDefense* approach consists of CNN, RNN (LSTM, GRU) and fully connected layers. The experimental results demonstrate that *DeepDefense* reduces error rate by 39.69% in *Data14* and from 7.517% to 2.103% in *Data15* compared with conventional machine learning method. Recurrent Neural Network can learn much longer historical features than conventional machine learning methods. Recurrent Neural Network also illustrates better performance in generalization than Random Forest does.

For future work, we plan to increase the diversity of DDoS vectors and system settings to test our model's robustness in different environments. Other shallow machine learning models will be included in the comparison. We will build a Deep DDoS Defense system based on *DeepDefense* and test it in the real world in the future.

Our current ISCX2012 dataset is provided by UNB in 2012.



(a) *Data14* Error Rate (b) *Data14* Accuracy (c) *Data14* Precision

(d) *Data14* Recall (e) *Data14* F1 Score (f) *Data14* AUC

Figure 5: Performance Comparison of LSTM and Random Forest (RF) model with different window sizes (*Data14*)

Table V: Performance Comparison of LSTM and Random Forest (Window Size 100) on *Data14* and *Data15*

ModelName	Error Rate	Accuracy	Precision	Recall	F1	AUC
LSTM	2.394%	97.606%	97.832%	97.378%	97.601%	99.096%
Random Forest	6.373%	93.627%	92.532%	94.893%	93.698%	93.628%

Many new features of DDoS attacks occur in recent years. For example, the attackers deliver larger but fewer packets to maximize the attack's bandwidth and intensity last year, according to a recent report from Akamai Technologies [37]. Other available public datasets are even older: KDD Cup 1999 [38], MIT Lincoln Laboratory DARPA 2000 [39], CAIDA DDoS Attack 2007 [40], and TUIDS DDoS dataset 2012 [41]. It is urgent to create a new dataset to meet the new challenges in DDoS identification.

REFERENCES

[1] V. D. Gligor, "A note on denial-of-service in operating systems," *IEEE Transactions on Software Engineering*, vol. 10, no. 3, pp. 320–324, 1984.

[2] L. D. Stein, *World Wide Web Security FAQ*. Lincoln D. Stein., 2002.

[3] P. J. Criscuolo, "Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319," DTIC Document, Tech. Rep., 2000.

[4] "Global application & network security report 2015-2016," Tech. Rep., 2016.

[5] "Q4 2015 state of the internet - security report," Tech. Rep.

[6] "Kaspersky ddos intelligence report for q4 2015," Tech. Rep., 2016.

[7] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and ddos problems," *ACM Computing Surveys (CSUR)*, vol. 39, no. 1, p. 3, 2007.

[8] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

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

[10] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[14] S. Specht and R. Lee, "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures," Technical Report CE-L2003-03, Princeton University, http://www. princeton. edu/˜ rblee/ELE572_ F04Readings. html, Tech. Rep., 2003.

[15] F. Guo, J. Chen, and T.-c. Chiueh, "Spoof detection for preventing dos attacks against dns servers," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE, 2006, pp. 37–37.

[16] J. Mirkovic, G. Prier, and P. Reiher, "Attacking ddos at the source," in *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*. IEEE, 2002, pp. 312–321.

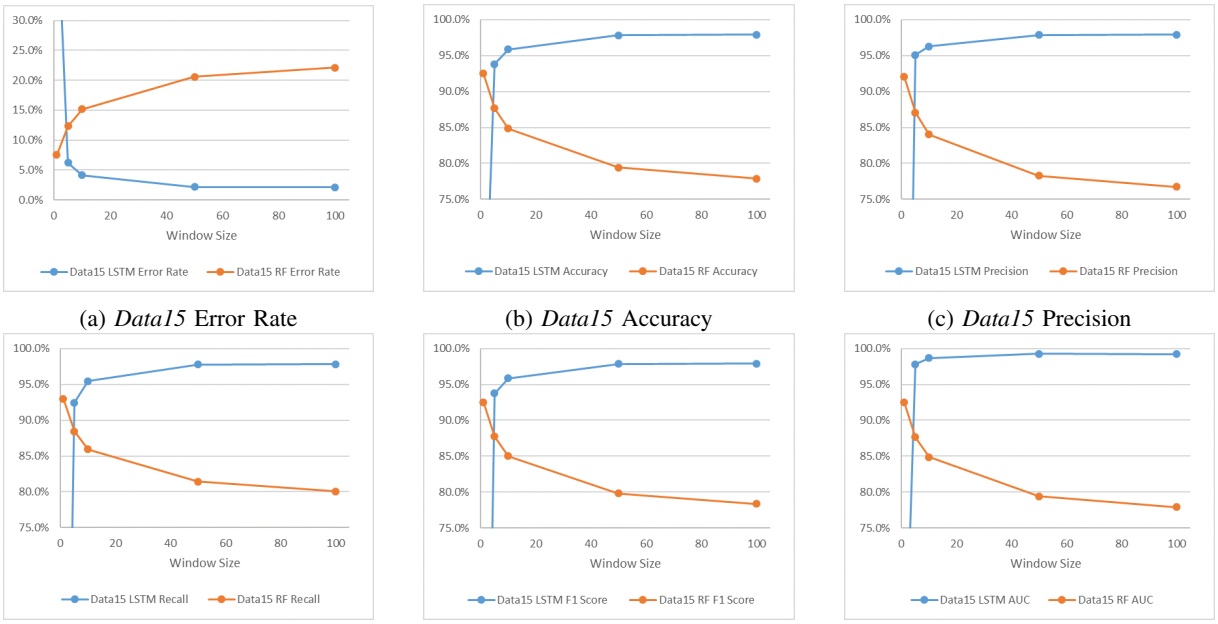
[17] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection," *Pattern Recognition Letters*, vol. 51, pp. 1–7, 2015.

[18] C.-L. Chen, "A new detection method for distributed denial-of-service attack traffic based on statistical test." *J. UCS*, vol. 15, no. 2, pp. 488–504, 2009.

[19] X. Xu, Y. Sun, and Z. Huang, "Defending ddos attacks using hidden markov models and cooperative reinforcement learning," in *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer, 2007, pp. 196–207.

[20] J. L. Berral, N. Poggi, J. Alonso, R. Gavalda, J. Torres, and M. Parashar, "Adaptive distributed mechanism against flooding network attacks based on machine learning," in *Proceedings of the 1st ACM workshop on Workshop on AISec*. ACM, 2008, pp. 43–50.

[21] T. Shon, Y. Kim, C. Lee, and J. Moon, "A machine learning framework for network anomaly detection using svm and ga," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. IEEE, 2005, pp. 176–183.



(a) *Data15* Error Rate (b) *Data15* Accuracy (c) *Data15* Precision

(d) *Data15* Recall (e) *Data15* F1 Score (f) *Data15* AUC

Figure 6: Performance Comparison of LSTM and Random Forest (RF) model with different window sizes (*Data15*)

[22] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: a statistical anomaly approach," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 76–82, 2002.

[23] S. Seufert and D. O'Brien, "Machine learning for automatic defence against distributed denial of service attacks," in *2007 IEEE International Conference on Communications*. IEEE, 2007, pp. 1217–1222.

[24] P. A. R. Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Computer Communications*, vol. 34, no. 11, pp. 1328–1341, 2011.

[25] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The ibm 2015 english conversational telephone speech recognition system," *arXiv preprint arXiv:1505.05899*, 2015.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[27] S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[29] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[31] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[32] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1113–1120.

[33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[35] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[36] W. Zaremba, "An empirical exploration of recurrent network architectures," 2015.

[37] B. Camarda, "Ddos attacks are soaring, says new report," Tech. Rep., 2016.

[38] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, vol. 2. IEEE, 2000, pp. 130–144.

[39] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Computer networks*, vol. 34, no. 4, pp. 579–595, 2000.

[40] U. CAIDA, "Ddos attack 2007 dataset," 2010. [Online]. Available: http://www.caida.org/data/passive/ddos-20070804dataset.xml

[41] P. Gogoi, M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "Packet and flow based network intrusion dataset," in *International Conference on Contemporary Computing*. Springer, 2012, pp. 322–334.