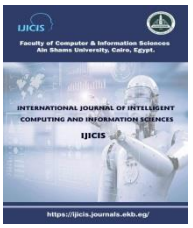
IJICIS, Vol.20, No.2, 125-138
DOI: 10.21608/ijicis.2021.51555.1046

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



Fast Detection of Distributed Denial of Service Attacks in VoIP Networks Using Convolutional Neural Networks

Waleed Nazih *
Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al Kharj, Saudi Arabia
w.nazeeh@psau.edu.sa

Yasser Hifny
Information Technology Department, Faculty of Computers and Information, Helwan University, Ain Helwan, Egypt;
yhifny@fci.helwan.edu.eg

Wail S. Elkilani
Computer Systems Department, Faculty of Computers and Information Sciences, Ain Shams University, Abassia, Egypt;
wail.elkilani@cis.asu.edu.eg

Tamer Abdelkader
Information Systems Department, Faculty of Computers and Information Sciences, Ain Shams University, Abassia, Egypt;
tammabde@cis.asu.edu.eg

Abstract: *Voice over Internet Protocol (VoIP) is a recent technology used to transfer media and voice over Internet Protocol (IP). Many organizations moved to VoIP services instead of the traditional telephone systems because of its low cost and variety of introduced services. The Session Initiation Protocol (SIP) is the most used protocol for signaling functions in VoIP networks. It has simple implantation but suffers from less protection against attacks. The Distributed Denial of Service (DDoS) attack is a dangerous attack that preventing legitimate users from using VoIP services and draining their resources. In this paper, we proposed an approach that utilizes deep learning to detect DDoS attacks. The proposed approach uses token embedding to improve the extracted features of SIP messages. Then, Convolutional Neural Network (CNN) was used to detect DDoS attacks with different intensities. Furthermore, a real VoIP dataset that contains different scenarios of attacks was used to evaluate the proposed approach. Our experiments find that the CNN model achieved a high F1 score (99-100%) as another deep learning approach that utilizes Recurrent Neural Network (RNN) but with less detection time. Also, it outperforms another system that depends on classical machine learning in case of low-rate DDoS attacks.*

* Corresponding author: Waleed Nazih

Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al Kharj, Saudi Arabia
E-mail address: w.nazeeh@psau.edu.sa

Keywords: *deep learning, convolutional neural networks, voice over IP, session initiation protocol, network security, distributed denial of service attacks*

1. Introduction

VoIP systems dependent on the infrastructure of IP network and use several protocols such as Real-Time Transport Protocol (RTP) [1] for voice and multimedia transferring and Session Initiation Protocol (SIP) [2] for communication between sessions. So, these systems are at risk of attacks of two sources; protocols of IP networks and its protocols [3]. Denial of Service (DoS) is an attack preventing legitimate users from using VoIP services. Malformed messages and flooding are the most frequent DoS attacks [4]. In the case of malformed messages attacks, the attacker sends a tampered SIP message to cause a partial failure or a restart of the SIP device when trying to process this message. Flooding attacks generate a large number of SIP messages to force the SIP device to consume resources such as memory. Therefore, the SIP device will probably be out of service for legitimate users [5]. To launch this attack, the attacker may use INVITE, REGISTER, and BYE methods. This attacker could be a legitimate user if such an attacker has an account in the SIP server or even an intruder who has violated the authentication requirements [6]. DDoS attacks are always produced by one computer that generates malicious traffic that seems to be coming from many sources using IP spoofing or by many computers managed by the attacker.

Furthermore, these attacks affecting work productivity and decreasing its earnings by breaking down one or more VoIP servers [7]. Discovering DDoS attacks is more complicated than DoS attacks because source blocking based on traffic limitations is useless. Furthermore, affected users are generally not aware that their systems are associated with a DDoS attack.

Machine learning techniques that were used to defend DDoS attacks [4] are introduced in section 2. The two main types of conventional machine learning techniques are supervised and unsupervised learning. Labeled data is the backbone of supervised learning. Preparing a labeled dataset usually expensive and time-consuming, so the lack of such datasets is the main bottleneck to supervised learning. On the other hand, unsupervised learning depends on extracting valuable features from unlabeled data which require deep knowledge of VoIP networks to choose these features.

Deep learning is a form of machine learning that simulates the human brain using multi-layer neural networks. Deep learning techniques don't require manual feature engineering, instead of this it directly learn feature representation from the original training data such as images and text. Therefore, it can perform in an end-to-end way. In addition, deep learning techniques possess a considerable advantage over machine learning techniques in the case of applications that have large datasets. Building a deep learning model usually involves choosing the model architecture (i.e., which deep learning technique suitable to a specific problem), optimization strategy, and hyper-parameters fine-tuning. Deep learning approaches have achieved significant results in many fields such as speech recognition [8], intrusion detection systems [9], and many other applications [10–14].

In this paper, an approach was proposed to detect DDoS attacks. This approach extracts features of each SIP message and feeds them into the CNN model to classify SIP messages as normal or malicious. Our contributions include the following: (1) Introducing DDoS detector using CNN; (2) Achieving high F1-score over low and high-rate DDoS attacks in addition to a low detection time; (3) Evaluating the proposed approach using a real balanced VoIP dataset; (4) Comparing our approach results with another deep learning approach and a conventional machine learning approach. The remainder of the paper is organized as follows. Section 2 introduces the previous work. Section 3 describes the proposed approach details. Section 4 explains the details of the dataset used for measuring system performance. Section 5 describes our experiments and results. Section 6 introduces our conclusions.

2. Related Work

Different approaches have been used for detecting DDoS attacks in VoIP networks. A change-point model was utilized in [15] to detect flooding attacks in DDoS form. The proposed work depends on monitoring changes in Mahalanobis distance between consecutive feature vectors in ten seconds intervals. If the distance exceeds a specific value, the system triggers an alarm. Moreover, the system calculated similarity scores over the behavior of users to discover attackers. A low accuracy was achieved and it has many parameters to be modified to capture traffic intensity. Kurt et al. [16] extracted more than 40 features from SIP messages and VoIP server measurements to detect DDoS attacks. A Hidden Markov Model (HMM) related the features to hidden variables, then a Bayesian multiple change model utilized them as an indication of an attack. In addition, a developed SIP
simulator and a commercial tool were used to generate normal messages and malicious messages respectively. Extracting a large number of features achieved a high detection accuracy but consumes memory and CPU resources.

Dassouki et al. [17] proposed an approach to discover flooding attacks. They utilized a state machine to create events for SIP sessions. Then, every SIP session is classified by a detector based on those events. Also, every session IP is stored in a database according to the detector decision. To evaluate the proposed system, they deployed an attack tool on many sites and developed low-rate and high-rate DDoS scenarios. Tas et al. [18] proposed a two-module approach for detecting advanced DDoS attacks. First, a statistics module calculates a group of thresholds over a specific duration of SIP traffic. Then, the second module used predefined rules to produce an action based on the previous thresholds. The proposed approach decreased the server CPU processing when it is under attack by 73.4%.

Support Vector Machine (SVM) was used to detect DoS attacks (i.e., INVITE flooding and malformed messages) and Spam over Internet Telephony (SPIT) in [19]. They converted SIP messages into n-grams tokens by moving a window over the SIP messages one by one. Then, the occurrences for each token is stored in the message feature vector. To classify the messages into normal or malicious, a fast linear SVM with l_1 regularizer was used. The proposed approach outperformed SVM in the dual form in terms of detection time and training time. Using deep learning approaches are rarely used to detect DoS and DDoS attacks [4]. Nazih et al. [9] proposed a DDoS detection approach based on RNN. An embedding layer was used to enhance the extracted features from SIP messages. The experiments showed that using

token-based achieved higher accuracy than character-based embedding. Besides, a real dataset was prepared using real SIP traffic from a VoIP network. Furthermore, a bidirectional RNN with Gated Recurrent Unit (GRU) achieved a high F1-score (i.e., 99%) and low detection time (i.e., 0.16 ms per SIP message).

3. Proposed Approach

The target of our proposed approach is to detect low and high-rate DDoS attacks in an efficient and timely manner. We used deep learning techniques to create and train a model that can automatically extract features to represent SIP messages with no need to feature engineering. This model was created in two phases, the first phase is the feature extraction process and the second phase is the creation of the CNN based model.

3.1. Feature Extraction

The feature extraction phase starts with the tokenizing process as shown in Figure 1. This process removes all punctuation from the SIP message and divides it into tokens using the space character as a delimiter. In addition, a dictionary that contains all unique tokens was created at the end of this process. Then, these tokens are converted to their equivalent indexes in the previous dictionary. To make all feature vectors have the same length, zeros were added at the end of them in the third process, which is called post padding.

In the last process of the feature extraction phase, we used the TensorFlow [20] token embedding layer to convert tokens vectors to continuous representation instead of its discrete form. This layer has an extra benefit of weights updating during model training, which may enhance the performance.

Figure 1. Feature extraction process

3.2. Deep Learning Model

The proposed deep learning model is based on CNN, which has distinct results in computer vision and image classification as a result of its capability to extract image features. The characteristics that made CNN useful in image processing makes them convenient for sequence processing [21].

CNN was designed to minimize the number of input data required for a traditional artificial neural network by using equal representation, sparse interaction, and parameters sharing [22]. Therefore, CNN is scalable and needs less time for the training process. CNN usually has two layers and the activation unit, as shown in Figure 2. The first layer (i.e., the convolutional layer) extracts features from input data using various kernels. The second layer (i.e., the pooling layer) enhances the generalizability of the features.

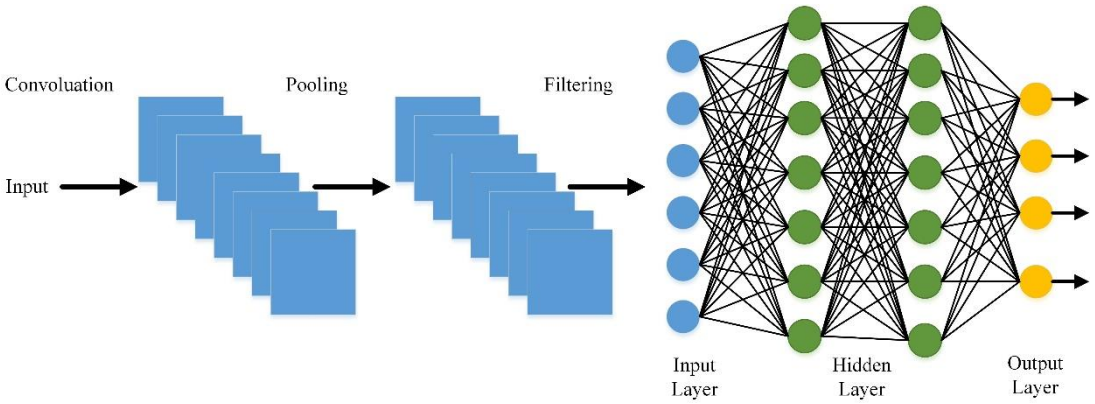


Figure 2. The architecture of CNN and its layers

In the case of dealing with images such as in image classification problems, images are handled as a two-dimensional matrix of numbers. Then, we detect features from these matrices which can be performed using CNN convolutional layers. Each one of these layers has a group of filters that are slid over the image to detect specific features.

When we are working with sequential data such as SIP messages, it is sufficient to use one-dimensional convolutions but with the same principle of picking up the patterns in the sequence. As shown in Figure 3, it takes a patch of features according to the size of the filter kernel. Then, it performs the dot product of this patch and the filter weights [23]. This one-dimensional convolution is invariant to translations, which enables CNN to recognizes specific sequences at different positions. This can be useful for certain patterns in the SIP message. The updates of the one-dimensional convolutional layer at each time step t are as follows:

$$\mathbf{h}_t^i = b_i + \sum_{j=1}^{d_x} \sum_{k=1}^{kw} \mathbf{W}_{ijk} \mathbf{x}_{dw(t-1)+k'}^j \qquad (1)$$

where the kernel width *kw*, the stride *dw*, and the output dimension $1 \leq i \leq dy$. $W \in R^{dy.dx.kw}$ and b are the free parameters.

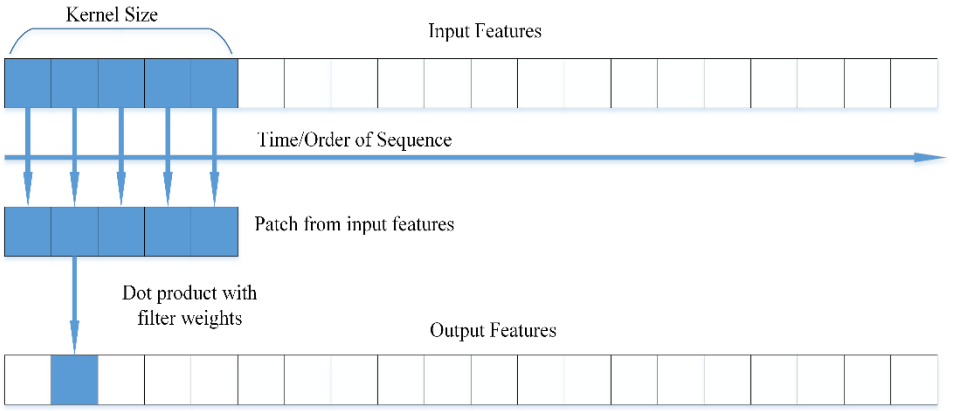


Figure 3. One dimensional convolution layer in CNN

The pooling layer of CNN is responsible for extracting a specific value from a set of values. The most common techniques used for this extraction is maximum pooling and average pooling. In this way, the size of the output matrix will be reduced since we are getting rid of unwanted information regarding the features existing in a specific part of the sequence data or the image. Furthermore, the CNN model's parameters will be reduced and less computation power will be required.

In our proposed approach, we used a CNN with one dimensional convolutional layer and one global pooling layer which converts the SIP message to one vector, as shown in Figure 4.

Deep learning models have a high probability of overfitting the training dataset, therefore we randomly select some layers' outputs and drop them during the training phase [24]. This is a powerful regularization technique that decreases overfitting possibility and increases the generalization of the model towards the unseen dataset. In addition, for the activation function of the model's hidden layer, a Rectified Linear Unit (ReLU) [25] was used. It is a fast and piecewise linear function that outputs the same input if it is positive and outputs zero otherwise. For the output of the proposed deep learning model, we were interested to determine the attack intensity (i.e., low or high). Hence, we used the softmax activation function in the output layer. It converting numbers into probabilities with sum equal to one and outputs probability distributions of a set of possible outputs as a vector.

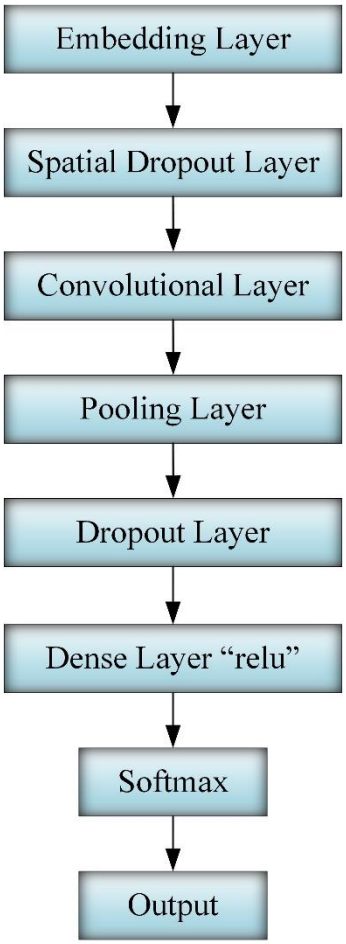


Figure 4. Full deep neural network model based on CNN

4. Dataset

To measure the performance of the proposed CNN model, we used a real balanced VoIP dataset [9]. The normal messages of this dataset were collected from a real VoIP network that has more than four thousand VoIP phones. In addition, it has many types of SIP messages such as INVITE, REGISTER, and ACK messages. To simulate the occurrence of DDoS attacks over the VoIP network, the SIPp-DD [26] generated the malicious messages that were injected into the dataset. One of the best advantages of SIPp-DD is its spoofing mechanism implementation to support the distributed attack generation. Since the most common and critical DDoS attack is the INVITE flooding [6], we adjusted the SIPp-DD to generate this attack.

Furthermore, this dataset contains three scenarios of DDoS attacks. The duration of these scenarios is 120, 60, and 30 s. Each scenario has five attack intensities; a very low attack using 10 messages/s, a low attack using 25 messages/s, a medium using 50 messages/s, a high attack using 100 messages/s, and a very high attack using 500 messages/s. We will refer to these attacks as VL, L, M, H, and VH
respectively.

This dataset consists of four datasets, a dataset for each scenario with six message classes (i.e., one class of normal messages and five classes of the previously mentioned attacks intensities) and the fourth dataset is the merging of the three scenarios datasets with 16 message classes to be used in the deployment of the CNN model. To avoid the imbalance between classes of every dataset, a tool was developed to mix the DDoS messages with the collected normal traces keeping an equal ratio between them. Besides, this tool divided every dataset randomly into three parts; training which is 60% of the whole dataset, validation and testing have an equal percentage of the dataset which is 20%. Table 1 summarizes the datasets' details.

Table 1. The details of the datasets

Scenario	Duration (s)	Number of Messages	Message Classes
Sc1	30	44,247	Normal, VH, H, M, L, and VL
Sc2	60	85,709	Normal, VH, H, M, L, and VL
Sc3	120	170,559	Normal, VH, H, M, L, and VL
Sc4	30, 60, and 120	300,515	Normal, Sc1:VH, Sc1:H, Sc1:M, . . . , Sc3:L, Sc3:VL

5. Experiments

We tested the proposed approach and measured its performance using the maximum and average pooling layer of CNN. In addition, we compared its detection accuracy and time with another deep learning approach based on RNN [9] and a classical machine learning approach based on SVM [19]. We implemented the proposed deep learning model using Python and TensorFlow [20] on a Tesla (R) P100 Graphics Processing Unit (GPU) and 24 GB memory at Google Colab.

5.1. Setup

The structure of the deep learning model and its hyper-parameters are strongly affecting its performance. Hence, we tried different combinations and values of them before settling on the structure shown in Figure 4 and the following values for its hyper-parameters: embedding layer dimension is 50, hidden layer units are 8, dropout value is 0.2, the learning rate is 0.001, and 100 epochs for the training.

Deep learning models are typically prone to overfitting. To prevent this as much as possible, we divided every dataset into three parts as mentioned before, and used them in the model training and testing as shown in Figure 5. During the training phase, the training part of the dataset was used to fine-tuning the hyper-parameters of the CNN model and the validation part measured the model performance after each epoch. After finishing the training phase, the testing part of the dataset, which is never used in the training phase was used to measure the model performance to produce the reported results.

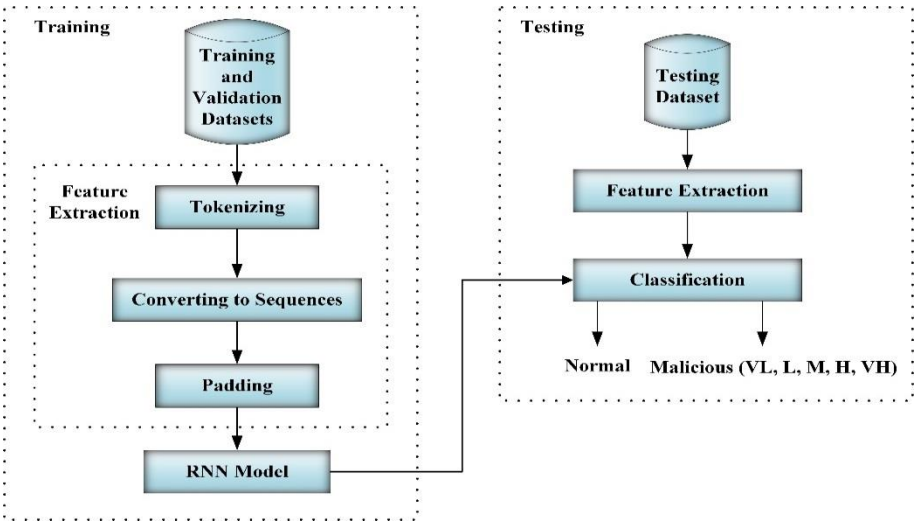


Figure 5. Training and testing workflow

The F1 score was used to measure the proposed approach performance. It is the harmonious average of the precision and recall that considers the false positives and false negatives [27]. Value of F1 score between zero and one, zero means that the model performance is too bad and one indicates a high performance of the model.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

$$\text{F1} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

For every experiment, we measured the F1 score and the loss of the training and validation parts of the dataset, the required time for training, the average detection time, and the confusion matrix to determine misclassified classes.

5.2. Results and Discussion

In our previous work [9], we found that extracting token-based features from SIP messages achieved high performance than character-based features. In addition, token-based is faster than character-based in training and detection. In the classification phase, we tried maximum pooling and average pooling with our proposed CNN model (i.e., CNN-MAX and CNN-AVG). In addition, we compared CNN results with the RNN model combined with GRU, which was reported to be faster than RNN combined with LSTM [9]. Furthermore, we compared CNN performance with a classical machine learning system based on SVM (i.e., l_1-SVM).

Table 2 introduces the results of the first experiment over DDoS scenarios. For the CNN model, we tried 32, 64, and 128 filters. Also, we changed the kernel size to 3, 5, and 7. The best results were achieved using 32 filters and 3 for the kernel size. The other values of the number of filters and kernel size didn't improve the accuracy or the detection time significantly. As shown in the table, CNN-AVG, CNN-MAX, and RNN-GUR almost detect different types of DDoS attacks, while the l_1-SVM didn't detect low-rate attacks.

Table 2. The performance of the proposed approach and other approaches.

Scenario	Classifier	F1 score	Misclassified Classes
Sc1	CNN-MAX	100%	nothing
Sc1	CNN-AVG	99.95%	nothing
Sc1	RNN-GRU	100%	nothing
Sc1	l_1-SVM	97.6%	L and VL

Sc2	CNN-MAX	99.99%	nothing
Sc2	CNN-AVG	99.98%	nothing
Sc2	RNN-GRU	99.9%	nothing
Sc2	l_1-SVM	97.5%	L and VL
Sc3	CNN-MAX	99.99%	nothing
Sc3	CNN-AVG	99.99%	nothing
Sc3	RNN-GRU	100%	nothing
Sc3	l_1-SVM	98.23%	L
Sc4	CNN-MAX	99.99%	nothing
Sc4	CNN-AVG	99.78%	Sc6: VL
Sc4	RNN-GRU	99.9%	nothing
Sc4	l_1-SVM	97.42%	Sc1: M, L, and VL - Sc2: L and VL - Sc3: L

Training and detection times are important factors that affect system deployment. In VoIP networks, the detection time should be as little as possible so as not to affect the network performance. Figure 6 introduces the required time for training in minutes, which is the total time of extraction of SIP features and classifier training. In addition, we compared the CNN training time with other state-of-the-art systems.

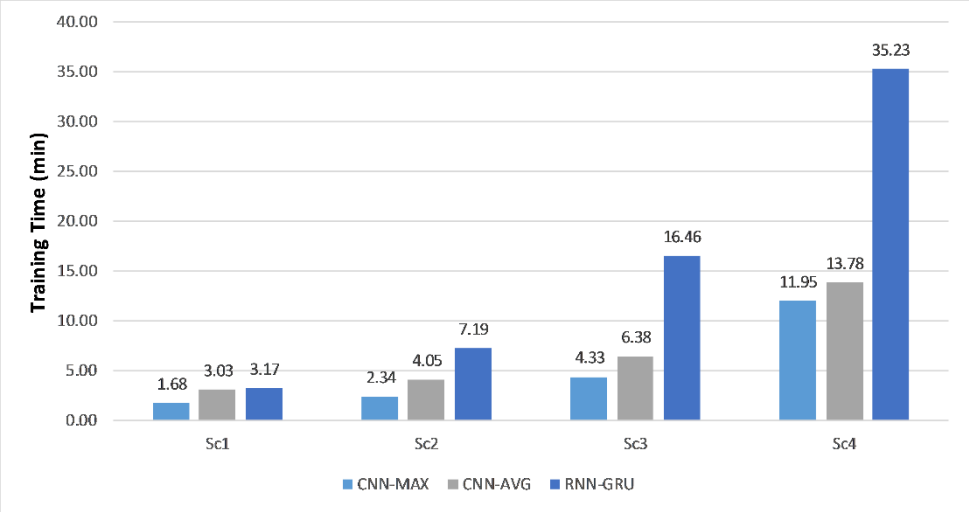


Figure 6. Training times of the CNN model and other approaches on a GPU

We used the testing part of the dataset to calculate the required time for detecting SIP messages in milliseconds. Figure 7 shows the detection time of the CNN model compared with other approaches. The detection time of CNN with maximum polling was comparable to CNN with average pooling. In addition, the detection time of the CNN model was about the third of the RNN-GRU detection time.

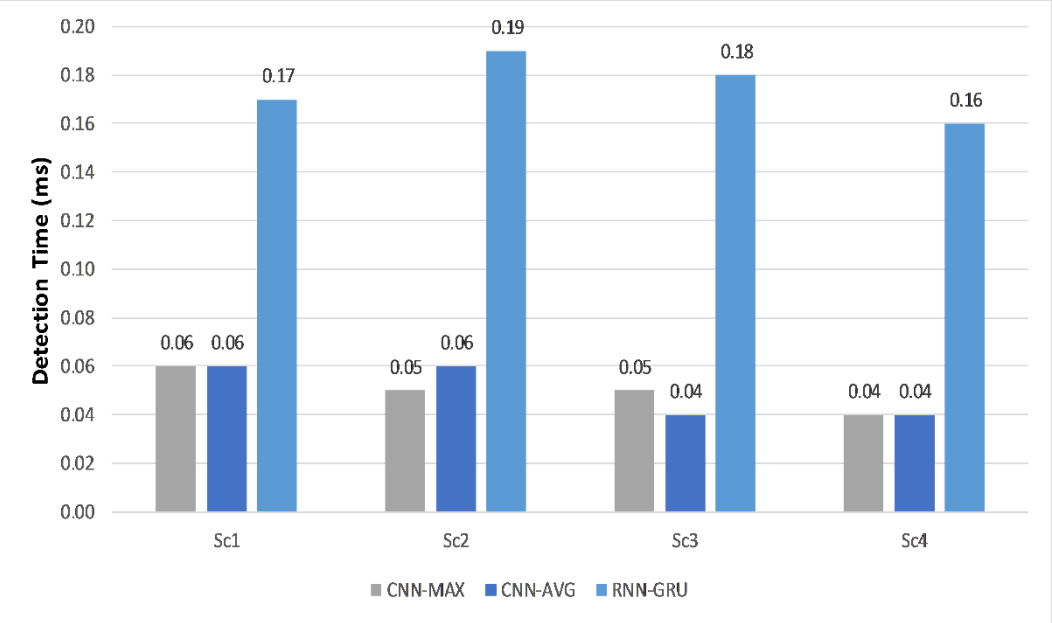


Figure 7. Detection times of the CNN model and other approaches on a GPU

6. Conclusions

Deep learning is considered state-of-the-art in many fields. We proposed a fast deep learning approach based on CNN to detect DDoS attacks. This approach converts SIP messages into features based on tokens, then used an embedding layer to enhances these features. The classification process depends on a CNN model. For the pooling layer of this model, we tried the maximum pooling and average pooling techniques. Our CNN model combined with maximum pooling achieved less training time. Concerning the required time of attack detection, it achieved comparable time to CNN with average pooling and a very little time compared to the RNN-GUR approach. Furthermore, it outperformed another approach that depends on SVM in detecting low-rate DDoS attacks. In the future, we will extend the proposed system to detect more SIP attacks such as multiattribute flooding.

References

1. Jacobson, V.; Frederick, R.; Casner, S.; Schulzrinne, H. RTP: A transport protocol for real-time applications. IETF RFC 3550 2003.
2. Rosenberg, J. SIP: Session Initiation Protocol. IETF RFC 3261 2002.
3. Keromytis, A.D. A comprehensive survey of voice over IP security research. IEEE Communications Surveys & Tutorials 2011, 14, 514–537.
4. Nazih, W.; Elkilani, W.S.; Dhahri, H.; Abdelkader, T. Survey of Countering DoS/DDoS Attacks on SIP Based VoIP Networks. Electronics 2020, 9, 1827.

5. Collier, M.; Endler, D. Hacking Exposed Unified Communications & VoIP Security Secrets & Solutions; McGraw-Hill Osborne Media, 2013.
6. Hussain, I.; Djahel, S.; Zhang, Z.; Nait-Abdesselam, F. A comprehensive study of flooding attack consequences and countermeasures in session initiation protocol (sip). Security and Communication Networks 2015, 8, 4436–4451.
7. Raza, N.; Rashid, I.; Awan, F.A. Security and management framework for an organization operating in cloud environment. Annals of Telecommunications 2017, 72, 325–333.
8. Saon, G.; Kuo, H.K.J.; Rennie, S.; Picheny, M. The IBM 2015 English conversational telephone speech recognition system. arXiv preprint arXiv:1505.05899 2015.
9. Nazih, W.; Hifny, Y.; Elkilani, W.S.; Dhahri, H.; Abdelkader, T. Countering DDoS Attacks in SIP Based VoIP Networks Using Recurrent Neural Networks. Sensors 2020, 20, 5875.
10. Ukaoha, K.C.; Igodan, E.C. Architecture Optimization Model for the Deep Neural Network. International Journal of Intelligent Computing and Information Sciences 2019, 19, 1–16.
11. Ukaoha, K.C.; Igodan, E.C. Architecture Optimization Model for the Deep Neural Network for Binary Classification Problems. International Journal of Intelligent Computing and Information Sciences 2020.
12. El-Ashmony, E.; El-Dosuky, M.; Elmougy, S. Classification of Low Quality Images Using Convolutional Neural Network and Deep Belief Network. International Journal of Intelligent Computing and Information Sciences 2016, 16, 19–28.
13. Habeeb, F.; Abuelenin, S.; Elmougy, S. Reducing Error Rate of Deep Learning Using Auto Encoder and Genetic Algorithms. International Journal of Intelligent Computing and Information Sciences 2016, 16, 41–53.
14. Al-furas, A.; AL-dosuky, M.; Hamza, T. Improving Feature Maps in Early Layers of Convolutional Neural Networks Using Otsu Method. International Journal of Intelligent Computing and Information Sciences 2016, 16, 37–45.
15. Semerci, M.; Cemgil, A.T.; Sankur, B. An intelligent cyber security system against DDoS attacks in SIP networks. Computer Networks 2018, 136, 137–154.
16. Kurt, B.; Yildiz, C.; Ceritli, T.Y.; Sankur, B.; Cemgil, A.T. A Bayesian change point model for detecting SIP-based DDoS attacks. Digital Signal Processing 2018, 77, 48–62.
17. Dassouki, K.; Safa, H.; Nassar, M.; Hijazi, A. Protecting from Cloud-based SIP flooding attacks by leveraging temporal and structural fingerprints. Computers & Security 2017, 70, 618–633.
18. Tas, I.M.; Ugurdogan, B.; Baktir, S. Novel session initiation protocol-based distributed denial-of-service attacks and effective defense strategies. Computers & Security 2016, 63, 29–44.
19. Nazih, W.; Hifny, Y.; Elkilani, W.; Abdelkader, T.; Faheem, H. Efficient Detection of Attacks in SIP Based VoIP Networks using Linear l_1-SVM Classifier. International Journal of Computers, Communications & Control 2019, 14.
20. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; others. Tensorflow: A system for large-scale machine learning. 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.
21. Abdel-Hamid, O.; Mohamed, A.r.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 2014, 22, 1533–1545.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.
23. Francois, C. Deep learning with Python, 2017.

24. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 2014, 15, 1929–1958.
25. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. ICML, 2010.
26. Stanek, J.; Kencl, L. SIPp-DD: SIP DDoS flood-attack simulation tool. 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN). IEEE, 2011, pp. 1–7.
27. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. Learning from imbalanced data sets; Springer, 2018.