Design and Implementation of Intrusion Detection System using Convolutional Neural Network for DoS Detection

Sinh-Ngoc Nguyen, Van-Quyet Nguyen, Jintae Choi, Kyungbaek Kim
Chonnam National University
Department of Computer Engineering
Gwangju, South Korea
sinhngoc.nguyen@gmail.com, quyetict@utehy.edu.vn, jefron1100@gmail.com, kyungbaekkim@jnu.ac.kr

ABSTRACT

Nowadays, network is one of the essential parts of life, and lots of primary activities are performed by using the network. Also, network security plays an important role in the administrator and monitors the operation of the system. The intrusion detection system (IDS) is a crucial module to detect and defend against the malicious traffics before the system is affected. This system can extract the information from the network system and quickly indicate the reaction which provides real-time protection for the protected system. However, detecting malicious traffics is very complicating because of their large quantity and variants. Also, the accuracy of detection and execution time are the challenges of some detection methods.In this paper, we propose an IDS platform based on convolutional neural network (CNN) called IDS-CNN to detect DoS attack. Experimental results show that our CNN based DoS detection obtains high accuracy at most 99.87%. Moreover, comparisons with other machine learning techniques including KNN, SVM, and Naïve Bayes demonstrate that our proposed method outperforms traditional ones.

CCS Concepts

• Computing methodologies→Probabilistic reasoning• Computing methodologies→Artificial intelligence

Keywords

Convolutional Neural Network, Machine Learning, Dos Detection, Network Traffic Formalization.

1. INTRODUCTION

In a complex network, the security is very essential for the operation of system. There are many kinds of techniques which are implemented in IDS to detect the DOS attack traffics such as misuse-based, anomaly-based method. Among those, misuse-based method is designed to detect known attack by using signature of those attack traffic[1].With other forms, the anomaly-based detection monitors the network traffic to find the deviation of normal traffic and attack traffic, which are employed to indicate the attack[2]. However, those techniques are only suitable for attacks which have been already known.

ICMLSC 2018, February 2–4, 2018, Phu Quoc Island, Viet Nam

ACM ISBN 978-1-4503-6336-5/18/02…$15.00
https://doi.org/10.1145/3184066.3184089

Nowadays, the rapid development of computer networks generates many security issues. The data exchangingfrom many sources, such as social network;online marketing; network applications,withthe leak security, can be collected by attackers.This information can be used by attacker to exploit other systems such as personal information database of government, private information in company[3]. Furthermore, the increasing of applications requires the services running on the server to adapt large amount of connections from users. This increment leads to more complicating network, which could possibly cause more security issues. Thereforeore, the servers can be easily exploited by attackers to be compromised, which then becoming the so-called botnets. These threats generatelot of attacks every year[5].

Currently, Machine Learning (ML)techniques such as KNN, SVM, Naïve Bayes are very popular. They are implemented in many fields, for example image processing, voice processing. Regarding to the IDS, there have been some researches applying ML to detect DoS attack [6][7].Thuy T.T. Nguyen and Grenville Armitage[6] survey the techniques of internet traffic classifications using ML. Benferhat et al. [7]proposed a Naïve Bayes approach to alert correlations. Their model needs a small part of expert knowledge and provides an efficient algorithm for detecting and predicting the plausible attack scenarios. However, there is another discriminative algorithm in ML that emerges as a well-known deep learning method in image processing, which is called convolutional neural network (CNN) model. It could be a suitable and potential technique to classify the network traffics and find the malicious ones. There are some researches in this area that apply the CNN to detect DoS[8][9].John Zhong Lei and Ali Ghorbani[8] provide a method based on competitive learning neural network to DoS attack that reduces the computation time. Deng, Chao, and Haiye Qiao[9]applied the basic CNN to detect the DoS attack. However, those researches are basic and primitive. It is not enough to show the efficiency of CNN in DoS detection. In this paper, we propose an IDS using CNN based method to detect DoS attack. And we provide an approach to normalize the network traffic to the input data of CNN, which is presented as a matrix of pixel. Also, we manipulate the designed model to get the configuration with the best result. Finally, we compare the performance of our model with other classifying methods such as KNN, SVM, Naïve Bayes.

The rest of paper is organized as follows. Next section is the background, which introduces the IDS, KDD dataset, and CNN model. In Section 3, we show the proposed IDS-CNN with CNN

based detection model. In this section, we also discuss about IDS-CNN architecture, how to normalize the input data for CNN, how to implement the IDS with CNN model. In Section 4, we conduct experiments to show the performance of our proposed method is suitable for DoS detection with the KDD dataset. Also, the comparison indicates that IDS based on CNN model provides a better accuracy and faster executing time than the other one.

2. BACKGROUND

2.1 Intrusion Detection System

Intrusion detection system is an important module in the network security system.Itis used to discover, determine, and identify unauthorized use, duplication, alteration, and destruction of information systems [10]. Most IDS focus on monitoring the operating system file and recognize the bad patterns such as malware to indicate the DoS attack. The others monitor the network traffic on the network devices, then they apply detection techniques to show the deviation of the attack and normal traffics. The traditional methods are suitable for only some specific attacks, but they cannot handle the diversity of network attack traffics in the system. Consequently, there is a need to have a method, which analyzes the characteristics and attributes of attack traffic, to be applied to the complex traffic.

2.2 KDD Cup 1999 Dataset

The KDD Cup 1999 dataset is a famous benchmarking for intrusion detection based on the 1998 DARPA initiative. It provides a benchmark to evaluate different methodologies. There are around 5 million records in KDD dataset. The dataset is made from 22 different attacks with 41featuresof traffic in each record. This achievement is made by the simulations running continuously in nine weeks on the local area network. The result is observed by three target machines running on various operating system and with different services. Then a sniffer records all network traffic using the TCP dump format. The attacks types are grouped into four categories:

- Denial of Service (DoS): Attacker tries to prevent legitimate users from using a service.
- Remote to Local (r2l): Attacker does not have an account on the victim machine, hence tries to gain access.
- User to Root (u2r): Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe: Attacker tries to gain information about the target host

2.3 Convolutional Neural Network Model

Convolutional neural network is a state-of-the-art model architecture for image classification tasks. CNN apply multiple filters to the raw pixel data of an image to extract and learn higher-level features, which the model can then use for classification. CNN contains three main components: input, output and hidden layers, which consist of convolutional layers, pooling layers, fully connected layers.

Firstly, convolutional layers apply a specified number of convolutional filters to the image. For each sub-region, the layer performs a set of mathematical operations with specific spatial extent and stride value to produce a single value in the output feature map. Convolutional layers then typically apply an activation function to the output to introduce nonlinearities into the model.

Secondly, the pooling layers scale down the image data extracted by the convolutional layers to reduce the dimensionality of the feature map in order to decrease processing time. In this layer, the commonly used pooling algorithm is max pooling, which extracts

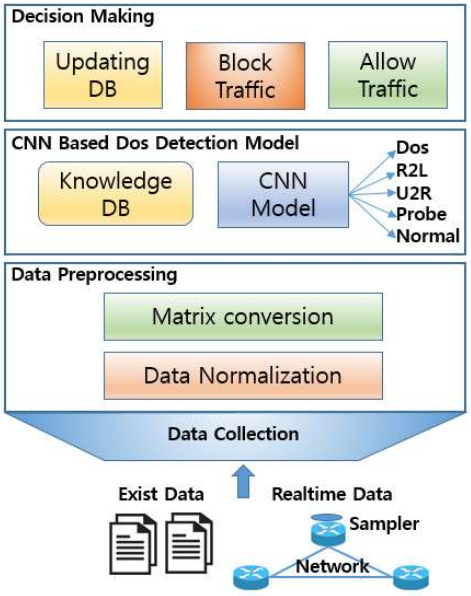

Figure 1. IDS-CNN Architecture

sub-regions of the feature map, keeps their maximum value, and discards all other values.

Lastly, fully connected layers perform classification on the features extracted by the convolutional layers and downscale bythe pooling layers. In a dense layer, every node in the layer is connected to every node in the preceding layer.

3. DESIGN AND IMPLEMENTIONOFIDS BASED ON CNN FOR DOS DETECTION

In this section, we introduce our proposed IDS based on CNN to detect Dos traffics. Then we explain how to normalize the data and generate the matrix pixel of data at the input data of CNN. We also discuss about how to implement CNN model to get the high detection performance.

3.1 Intrusion Detection System Architecture

Our goal is to design and implement anIDS for Dos detection. In this work, we propose an IDS-CNN architecture to detect DoS attack traffic including 4 main layers such as Data Collection, Data Preprocessing, CNN based DoSDetection Model and Decision Making which shows in Figure 1.

Data Collection: The lowest layer in the system, it receives the real-time network traffic from sampler or collector system as well as the existing data such as KDD Cup 99.

Data Preprocessing: This step will prepare the input data for CNN model from the raw data. After data is collected in the lower layer, they will be pre-processed by data normalize module. These data may include some very big value, small value, and different type. It needs to be normalize into one type and in a range [0…255]. And matrix conversion module will convert data into the matrix as the input of CNN model after we normalize it.

CNN based Dos Detection Model: The next layer is detection model, which uses trained CNN model to classify the input trafficinto five types of traffic, which could be normal traffic or four kinds of attack traffics as mentioned before. The accuracy of

detection depends on the knowledge. We provide the knowledge database for CNNtraining.

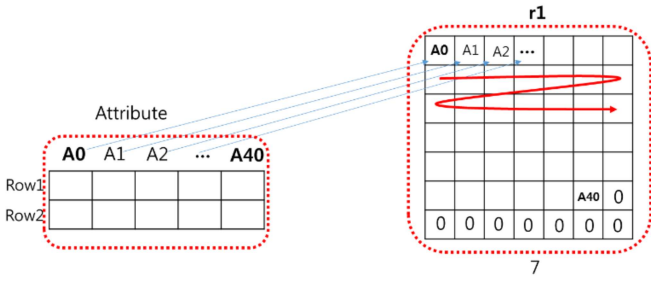

Figure 2. KDD Database Normalization

Algorithm 1: Dataset formalization
Require: KDD Dataset
Ensure: New formalized data with range from 0 to 255
1. c = **foreachColumn**()
2. avg = 0;
3. r = 0
4. **if**(c is String) then do
5. new_val=**ProcessString**(c)
6. **else**
7. avg = **average**(c)
8. r = **getRow**(c)
9. **if**(r < 122) **then do**
10. new_val = r * 2
11. **else**
12. **if**(r < 2 * avg) **then do**
13. new_val = (r * 123) / avg
14. **else**
15. new_val = 255

Decision Making: The last layer in this architecture is decision making, it is used to give the policies to the traffic after we get the result. After having the result of classification, a decision should be given to block or allow the traffic. If that traffic is detected as attack, we can block it or reroute to other server for more analysis. Furthermore, we can use the detected result to update the knowledge database to increase the detection ability of the system.

3.2 Data Normalization

CNN is proposed as a well-known image classification model, so the input format of CNN should be image. However, in some cases, the CNN can also be applied to classify the voice or text, and the input data are quite different. Therefore, those kinds of datasets should be normalized into the general one, which is a matrix containing the value of the pixelof the image. Each value in a pixel has the value from 0 to 255.

The KDD 1999 dataset is the network traffic that includes 41 attributes in a record with heterogenous of types. It may bea string or integer or float, and the range is quite different. It is not suitable to input this dataset into CNN, we had to normalize the data into a new dataset that contains the integer value, and the range is from 0 to 255. Also, by the intuition, we can see that most of the values in the KDD dataset are lower than 122 and there area few some abnormal cases being larger than 255.To normalize the dataset, we provide ***Algorithm 1*** to preprocess the data. If the data is astring, we can easily match the value to aninteger from 0 to 255 at line 5. In the complex case, we need to calculate the average value for each column, which could be shown at line 7 of ***Algorithm 1***. Then, if the value in any row less than 122, we can normalize them by doubling the old value. Otherwise, if the old

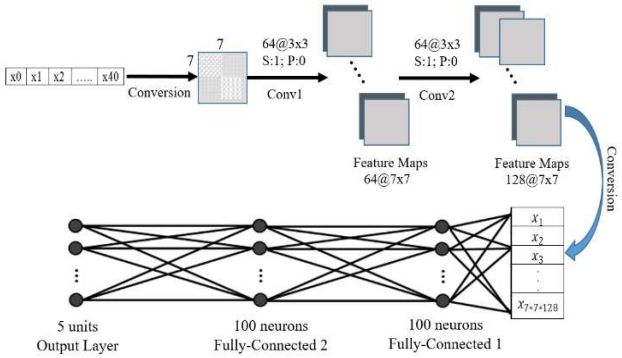

Figure 3. Dos Detection based on CNN Model

value is less than the double of average, we can calculate the new value at line 13. The other case that out of 255, we assign 255 for the new value.

After having a new dataset, there is a question: how can we present them by matrix of pixel. To answer this question, we think about the input image of CNN, it should be a square matrix, and the most suitable smallest one to contain 41 attributes is a 7x7 sized matrix. We can convert each record in the new dataset into a square matrix with 7x7 of size which showed in Figure 2. For the last eight bits of the matrix, we set them to 0.

3.3 Implementation of CNN for Dos Detection

Generally, there is no specific CNN architecture used to classify the input data with the best performance, so we design and implement the architecture with different configurations, then execute it several times to decide the best one. As we have described above, a simple CNN architecture is a series of triple layers, each of them transforms one volume of activations to another through a differentiable function. In our system, we design a CNN architecture with two *Convolutional Layers* and three Fully-Connected Layers as shown in Figure 3. The *Pooling Layers* are not used in our CNN architecture because it is unnecessary to perform a down-size operation for samples having very small size (e.g., the image size is 7x7) in our dataset. In more detail:

Convolutional Layers: two convolutional layers are used in our design. In the first convolutional layer, *Conv1*, we design using 64 filters with [3x3] of size. The input data for this layer is 2D images which are generated from network traffics, each has the size [7x7]. This layer will result in feature maps with size [7x7x64]. The second convolutional layer, *Conv2,* use 128 filters whose size is [3x3], so it results in feature maps with size [7x7x128]. Both of two convolutional layers used the same parameters: stride, S=1; zero-padding, P=0; ReLU activation function which leaves the size of the volume unchanged in each layer.

Fully-Connected Layers: we use three fully-connected layers. The feature maps which are generated by *Conv2* are used as the inputs for the first fully-connected layer. The second fully-connected layer uses the same parameters as the first one: the number of hidden units, h=100; bias b=0; and activation function is ReLU. The last fully-connected layer (Output Layer), will compute the class scores, resulting in the volume of size [1x1x5], where each of the 5 numbers corresponds to a class score, such as among the

5 categories of DoS attacks. Besides, we use drop-out parameter, d = 0.5, to avoid overfitting problem during CNN training phase.

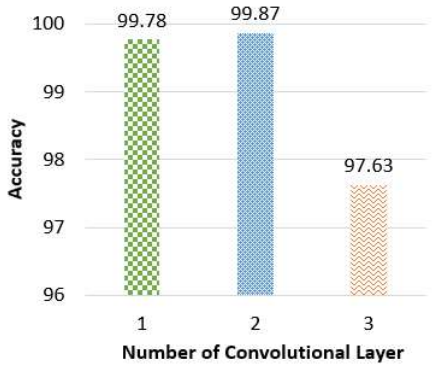

Figure 4. Change Number of Convolution

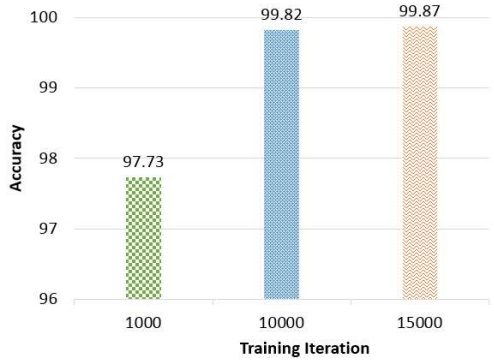

Figure 5. Change Training Iteration

4. EVALUATION

4.1 Evaluation Setting

To evaluate our proposed model, we conduct an experiment and show the performance of our proposed CNN model. Then we compare the performance of CNN model and other machine learning techniques. In this experiment, we employ TensorFlow library with Python language to implement the CNN model. We evaluate CNN model with two different settings. First experiment, we change the number of convolutional layer of the model from 1 to 3 layers. Then we keep the number of layer that gives best result and change the number of training iteration with 1000, 10000, and 15000 of training iteration. After that, with the configuration that gives the best result, we compare the accuracy of CNN to other machine learning techniques such as SVM, Naïve Bayes and KNN. Also, we compare the execution time each of those methods.

Regarding the input dataset, we use the first 1 million records in KDD dataset. 70% of them will be used for training and 30% for testing. We implement Algorithm 1 to preprocess data, and convert the data for each record into the matrix 7x7 as the input data. Then we apply CNN model for Dos attack classification.

4.2 Evaluation Result

In our experiment, we compare the result of detection when we change the number of convolutional layer from 1 to 3 layers. The result of detection in this change is shown in Figure 4. The accuracy increases to 99.87% when we increase the convolution up to 2 layers. However, it decreases to 97,63 when we use 3 convolution layers.

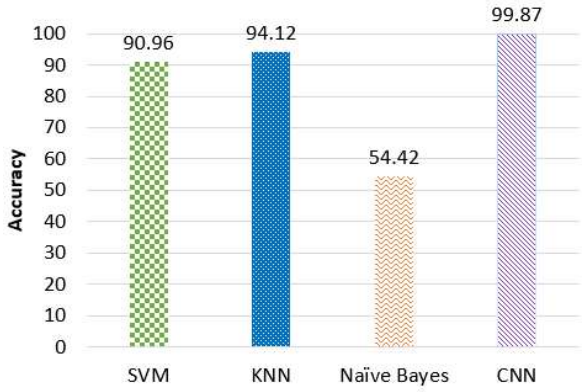

Figure 6. Accuracy of CNN and other ML Techniques

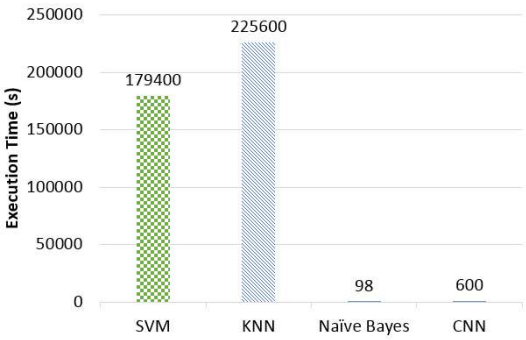

Figure 7. Execution Time of CNN and other ML Techniques

In the second test, we change the training iterations value with 1000, 1000 and 15000. The best result gives at 15000 training iterations with 99.87% which showed in Figure 5. If we increase the number of training iteration, we can increase the accuracy.

However, it spends much time to execute the training. By several changing in the model, we get the best configuration that provides the best result in performance with 2 convolutional layers and 15000 training iteration.

After that, we apply this configuration to compare the performance of CNN model with other ML techniques such as KNN, SVM, Naïve Bayes. We compare the accuracy and show the result in Figure 6. In this comparison result, the CNN reaches highest accuracy of detection with 99.87%, and Naïve Bayes gives lowest detection with 54,42%. However, the execution time of Naïve Bayes is smallest, it spends 98 seconds for execution, CNN model spends 600 seconds, and KNN spends much time with 225600 seconds. The result is shown in Figure 7.

With our proposed model and the evaluation to show the performance of Dos detection with KDD dataset, CNN can be the best in this situation. It is suitable to detect DoS attack in KDD dataset.

5. CONCLUSION

In this paper, we propose a IDS-CNN platform to solve the challenge of Dos detection. The other ML techniques spend much

time for detection or give a low accuracy. Because, there is not the best model for specific dataset, we had to run with many different configurations to get the best one. Also, we provide an approach to normalize the raw data for input data of CNN model. The conducted experiment shows the performance of each change in CNN model and comparison of performance between CNN model and other ML classification techniques.

6. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2017R1A2B4012559)

7. REFERENCES

[1] A. S. a. D. P. G. Desai, "Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA," *Advances in Electronics, Communication and Computer Technology (ICAECCT),* pp. IEEE International Conference on. IEEE, 2016., 2016.

[2] P. e. a. Garcia-Teodoro, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security,* vol. 28.1, pp. 18-28, 2009.

[3] G.-J. M. S. a. A. S. Ahn, "Security and privacy in social networks," *IEEE Internet Computing ,* vol. 15.3, pp. 10-12., 2011.

[4] G. e. a. Kulkarni, "Cloud security challenges," *Telecommunication Systems, Services, and Applications (TSSA),* vol. 2012 7th International Conference on. IEEE, 2012.

[5] C. e. a. Kolias, "DDoS in the IoT: Mirai and other botnets," *Computer ,* vol. 50.7, pp. 80-84, 2017.

[6] T. T. a. G. A. Nguyen, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials ,* vol. 10.4, pp. 56-76, 2008.

[7] S. T. K. a. A. M. Benferhat, "A naive bayes approach for detecting coordinated attacks," *Computer Software and Applications,* Vols. 32nd Annual IEEE International. IEEE, 2008, p. COMPSAC'08, 2008.

[8] J. Z. a. A. G. Lei, "Network intrusion detection using an improved competitive learning neural network," *Communication Networks and Services Research,* Vols. Second Annual Conference on. IEEE, 2004., p. Proceedings, 2004.

[9] C. a. H. Q. Deng, "Network security intrusion detection system based on incremental improved convolutional neural network model," *Communication and Electronics Systems (ICCES),* pp. International Conference on. IEEE, 2016., 2016.

[10] S. A. S. a. A. A. Mukkamala, "Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools," *Vemuri, V. Rao, Enhancing Computer Security with Smart Technology.,* pp. (2005): 125-163., (Auerbach, 2006) .