INTRUSION DETECTION SYSTEM USING DEEP LEARNING FOR DoS ATTACK DETECTION

[1]ANDRE ARTA KURNIAWAN, [2]JUSAK, [3]MUSAYYANAH

Program Studi/Jurusan Teknik Komputer, University Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

e-mail: [1]andrearta1@gmail.com, [2]jusak@dinamika.ac.id, [3]musayyanah@dinamika.ac.id

ABSTRACT

Various attacks on a computer network or the internet have generated many incidents and cases, this makes security threats in using the internet or computer networks a major focus. Denial of Service attack or often referred to as DoS attack is one of the attack techniques that carry out flooding packets or requests to the target computer until the target computer is down. Prevention is needed in order to minimize existing attacks. IDS can be used as a detector in network traffic, but because IDS has its limitations, an IDS system is built using Deep Learning to detect DoS attacks. By using the data from the wireshark log as a dataset, it is necessary to do data normalization which will then be inputted into CNN VGG-19. The test results that have been carried out with variations in the data inputted into the CNN VGG-19 produce an average accuracy of 99.32% with an average loss of 4.08%, and by varying the iteration of the training process the resulting accuracy is 99.17% with an average loss - an average of 4.46%. And the ROC Curve value for the True Positive Rate and the False Positive Rate is 1.

Keywords: *Intrusion Detection System, DoS Attack, Deep Learning, Convolutional Neural Network, Wireshark*

1. INTRODUCTION

In the present day, there is a lot of case has been founded such as internet users who are under digital attacked. (for example, "Bobol Akun e-Banking, Pemilik Kehilangan Uang Rp1 Miliar" (Abdi, 2019)). There are also cases of suddenly the user's smartphone locks by itself (for example, "Hati-hati Pakai WiFi Gratisan, HP Bisa Diretas Hacker Jahat" (Franedya, 2019)).

Digital attacks that threaten internet security are a type of Denial of Service (DoS) attack. In general case, DoS attacks carried out with "flooding" the computer network with a many requests at the same time. The purpose of these attacks are make the computer network down so that the other users unable to used the current connection in their computer network system. (Dogru & Masetic, 2017).

One way to solve these kind of network attacks is to use an Intrusion Detection System (IDS), which has function to detect if there are attacks into a server. IDS works by observing for every traffic that exist on network and will give a warn if there is abnormal traffic

Therefore, IDS is very important in computer networks, there are a lot of research about computer networks that focuses on developing IDS. One of them is by applying machine learning method on IDS. (Zamani & Movahedi, 2013). Machine learning is one of branches computer science which give a computer system capabilities to learn from the data without explicitly programmed. (Purnama, 2019).

The deep learning method is one of the developments of machine learning that studies data more deeply and uses a multi-layered learning process. This study will apply one of deep learning methods, also known as the Convolutional Neural Network algorithm which will increase the capability of IDS to detect of attacks. Because attack traffic often has multiple models, IDS with CNN can study the traffic itself without writing an explicit detection program.

DOI: http://doi.org/10.54732/jeecs.v6i2.203

Previous IDS research analyzed IDS capabilities using the KDD 1999 dataset (Sinh-Ngoc Nguyen dkk, 2018). However, this research will use SYN Flood, UDP Flood, and Ping of Death attack types to test CNN's ability on IDS. And also, this study will also use log results from Wireshark and the ISCX-IDS-2012 dataset as training data.

2. METHOD

The Deep Learning method can be considered as a combination of Machine Learning with Artificial Neural Networks (Primartha, 2018). In deep learning, computers study from the data using a layered network of neurons. This is similar to the structure of the human brain, which studies sensor data over a neuron network that so complex.

In this study, the CNN deep learning method was used to detect attacks on network traffic. The system design used is shown in Figure 1.

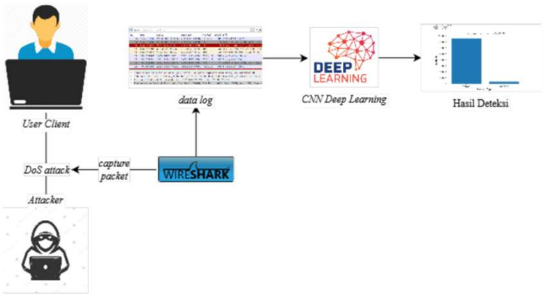


Figure 1. IDS System Design Using Deep Learning

(Source: Kurniawan, 2020)

All network traffic data will be captured by the Wireshark program, a program that can capture all packets crossing the network. From the captured packet, log data is used as CNN input data.

Because CNN is a branch of machine learning, a training process is needed before CNN can be used to test data. For the IDS system training process, various log data are made with a mixture of normal traffic data and attack data. After the training process is complete, the system can be validated with test traffic data.

2.1 Convolutional Neural Network

The CNN architecture is similar to the visual cortex of an animal, in that the neural network receives input image data. The input data for each CNN layer also uses a three-dimensional structure.

In generally, the CNN architecture consists of input layers, output layers, and hidden layers. The hidden layer on CNN is often referred as the Feature Extraction Layer (Sena, 2017). And the hidden layer consists of a Convolution Layer and a Pooling Layer.

At the Convolution Layer, the dimensions of the input data will be taken according to the filter size. For example, if the filter dimension size is 3x3 and the input data dimension size is 5x5, then the input data dimension will be taken according to the filter size which is 3x3. Then the multiplication process will be carried out as many as the number of filters used.

The Pooling Layer functions are to maintain the size of the data when convolution is carried out, namely by reducing the sample or also known as downsampling (Suyanto, Ramadhani, & Mandala, 2019). Max pooling is the most common polling process, while the average pooling method is rarely used (Michelucci, 2019).

The CNN architecture in this study uses the VGG-19 model. The architecture consists of 16 convolutional layers, 5 pooling layers, and 3 fully-connected layers. CNN architecture as a whole, there are 24 layers and 19 layers that have weight, because the pooling layer has no weight. In the fully connected layer, each neuron has full connection to all activations in the previous layer.

The convolutional layer used in this study has a different number of filters. Conv-1 has 64 filters, Conv-2 has 128 filters, Conv-3 has 256 filters, while Conv-4 and Conv-5 have 512 filters.

In the VGG-19 architecture, the activation layer used is ReLU or Rectified Linear Unit. The ULT function will apply the maximum activation function f (x) = max (0, x). If the value of x is negative, then its function will set the return value to zero. Meanwhile, if the value of x is positive, then its function will set the return value in accordance with x.

The loss function used in this study is binary classification. This is because the IDS system functions to detect only two statuses or two classes, namely DoS attack conditions or normal network conditions.

2.2 DoS Attack

The types of DoS attacks that are simulated in this study are SYN Flood, UDP Flood, and Ping of Death, which are the most common types of attacks in 2015. In general, a DoS attack sends many request packets to make the computer network down.

In a SYN Flood attack, the attacker sends multiple SYN packets so the target is forced to continue responding to these packets by repeatedly sending SYN ACK packets. In reality, when an attacker uses a SYN Flood it is only intended to overrun the target and does not intend to establish a connection using the Transmission Control Protocol (TCP).

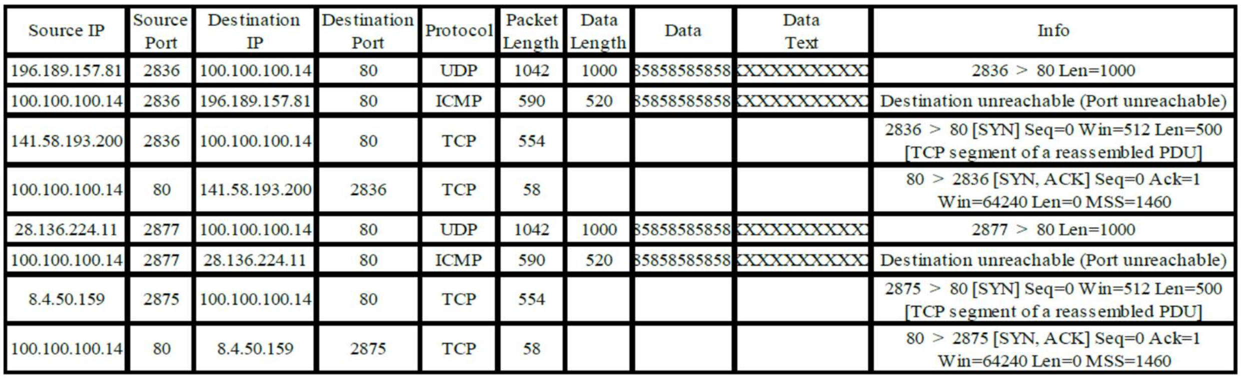
In a UDP Flood attack, the attacker takes advantage of the User Datagram Protocol (UDP) characteristic that can directly send packets to the target without the need to make a connection like TCP. With the large amount of data sent by the attacker suddenly, it makes the target down.

In a Ping of Death attack, the attacker sends many ping packets to the target so that the target becomes overwhelmed and has to reply to all of the ping packets. Although generally the length of a ping packet is only 32 bytes, but the ping packet size can be set up to a maximum of 65 KB. With such a large number of ping packet traffic in a DoS attack, the network can crash.

2.3 Dataset

Examples of log data from Wireshark can be seen in Table 1. There are 10 packet parameters used as input on CNN, namely the source IP address, source port, destination IP address, destination port, protocol, packet length, data length, data in hexadecimal form, text data, and package information.

Table 1. Log Data for CNN Input



Source IP	Source Port	Destination IP	Destination Port	Protocol	Packet Length	Data Length	Data	Data Text	Info
196.189.157.81	2836	100.100.100.14	80	UDP	1042	1000	85858585858	XXXXXXXXXX	2836 > 80 Len=1000
100.100.100.14	2836	196.189.157.81	80	ICMP	590	520	85858585858	XXXXXXXXXX	Destination unreachable (Port unreachable)
141.58.193.200	2836	100.100.100.14	80	TCP	554				2836 > 80 [SYN] Seq=0 Win=512 Len=500 [TCP segment of a reassembled PDU]
100.100.100.14	80	141.58.193.200	2836	TCP	58				80 > 2836 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
28.136.224.11	2877	100.100.100.14	80	UDP	1042	1000	85858585858	XXXXXXXXXX	2877 > 80 Len=1000
100.100.100.14	2877	28.136.224.11	80	ICMP	590	520	85858585858	XXXXXXXXXX	Destination unreachable (Port unreachable)
8.4.50.159	2875	100.100.100.14	80	TCP	554				2875 > 80 [SYN] Seq=0 Win=512 Len=500 [TCP segment of a reassembled PDU]
100.100.100.14	80	8.4.50.159	2875	TCP	58				80 > 2875 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

(Source: Kurniawan, 2020)

From all the Wireshark log results, the data will be divided into two parts, 80% for training data and 20% for test data. And also, in the training process, for each variation of the number of iterations will be ran for 10 times, ranging from 10 iterations to 100 iterations with multiples of 10.

2.4 Data Normalization

Because the CNN method is generally used for image input data, the Wireshark log data in the form of text must first be determined using the ASCII code so that it becomes a value between 0 and 255. This corresponds to the color value in an image pixel, expressed in bytes.

After the data is converted into byte values, the data is arranged into a 32x32x3 three-dimensional matrix. Size 3 in the third dimension resembles the number of layers in a color image, which consists of layers of red, green, and blue. While the size 32x32 is intended to form an image with a size of 1024 pixels.

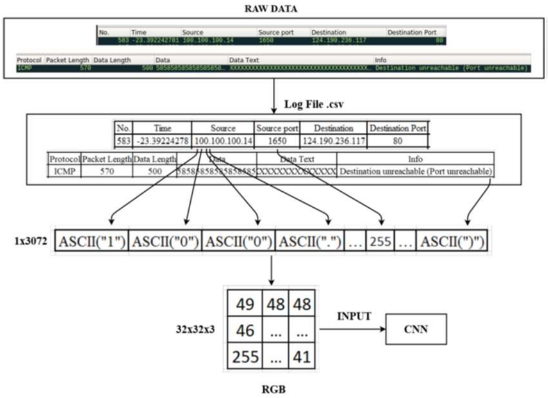

RAW DATA
No. Time Source Source port Destination Destination Port
583 -23.392242781 100.100.100.14 1650 124.190.236.117 80
Protocol Packet Length Data Length Data Data Text Info
ICMP 570 500 585858585858585858… XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX… Destination unreachable (Port unreachable)
Log File .csv
No. Time Source Source port Destination Destination Port
583 -23.39224278 100.100.100.14 1650 124.190.236.117 80
Protocol Packet Length Data Length Data Data Text Info
ICMP 570 500 85858585858585858 XXXXXXXXXXXXXXX Destination unreachable (Port unreachable)
1x3072
ASCII("1") ASCII("0") ASCII("0") ASCII(".") … 255 … ASCII(")")
32x32x3
49 48 48
46 … …
255 … 41
RGB
INPUT
CNN

Figure 2. Normalization of Data for CNN Input

(Source : Kurniawan, 2020)

If the number of pixels in the CNN input image is counted, then the number will require 3072 data. When the data for one packet from the Wireshark log is less than 3072 bytes of data, then the data will be duplicated until one packet can meet the 3072 pixel images needed as one CNN input data.

2.5 Accuracy Calculation

This attack detection system is a system that only decides a binary value, so testing the IDS system can be done by calculating the True Positive Rate (TPR) value and the False Positive Rate (FPR) value. From the TPR and FPR values, the level of accuracy of the attack detection system using the CNN method can be analyzed.

The TPR value is obtained from equation 1. The True Positive (TP) value is obtained from the number of DoS attack data that is correctly detected as an attack, while the False Negative (FN) value is the amount of DoS attack data that is incorrectly detected as normal traffic.

$$TPR = TP/(TP + FN) \qquad (1)$$

The FPR value is obtained from equation 2. The False Positive (FP) value is obtained from the amount of normal traffic data that is incorrectly detected as an attack, while the True Negative (TN) value is the amount of normal traffic data that is correctly detected as not an attack.

$$FPR = FP/(FP + TN) \qquad (2)$$

3. RESULT AND DISCUSSION

3.1 CNN Accuracy Testing Against Data Variations

The purpose of this test is to test how accurate the CNN system is in detecting DoS attacks with varying data. The variation data used is taken from the results of capturing the wireshark log which is done 5 times with the wireshark capturing for about 5 minutes. From the log results, different traffic logs are obtained with different amounts of data. This log data will then be normalized first which will then be inputted into CNN.

Table 2. The Results of Testing The Accuracy of CNN on Data Variations

No	Total Data	Network Activity		Loss (%)	Accuracy (%)
		Normal (%)	DoS (%)		
1	74926	62.72%	37.28%	4.26%	99.27%
2	59934	58.42%	41.58%	4.11%	99.24%
3	50981	63.26%	36.74%	5.76%	99.33%
4	30995	69.96%	30.04%	2.96%	99.55%
5	433375	68.10%	31.90%	3.30%	99.20%

(Source: Kurniawan, 2020)

Table 2 shows the results of the CNN test on data variations. From table 2 the test results can be seen that the total amount of data generated when the wireshark captures varies with the same duration for approximately 5 minutes. The resulting percentage for normal network activity averaged 64.49% and the average DoS attack activity percentage was 35.50%.

In testing, a training process was carried out with a total of 50 iterations for all data, and the smallest loss was 2.96% and the greatest was 5.76% and the accuracy results were 99% for all data testing.

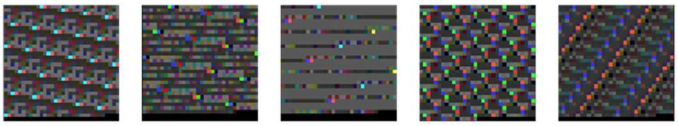


Figure 3. The Results of Normalization of Test Data are Plotted Into Pictures.
(Source: Kurniawan, 2020)

Figure 3 shows the results of the data that have gone through the data normalization process and then plotted into a picture. This is done because the neural network used in this Final Project research uses images as input.

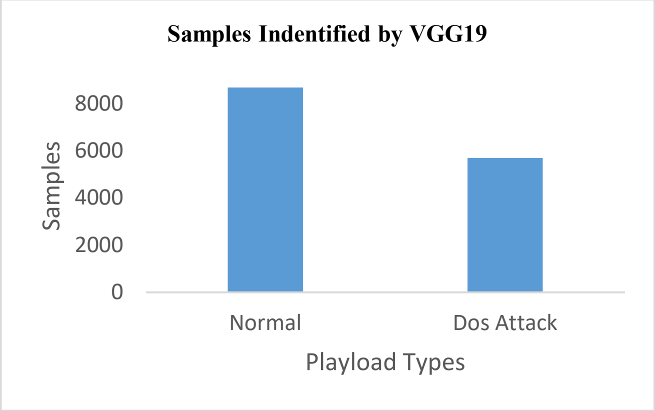


Figure 4. Network Activity Bar Graph
(Sumber: Kurniawan, 2020)

Figure 4 shows the bar graph of the CNN detection results between DoS attack activity and non-DoS attack. The data used in this detection uses separate test data from training data.

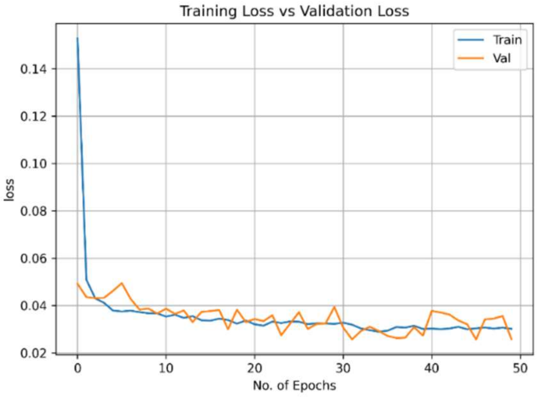


Figure 5. Loss Chart of The Training Process And The Validation Process With 50 Iterations
(Sumber: Kurniawan, 2020)

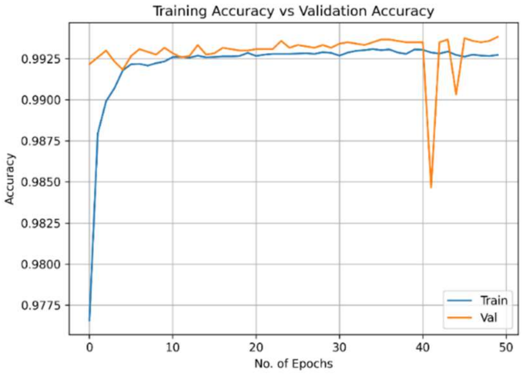


Figure 6. Graph Accuracy Training Process And Validation Process With 50 Iterations

(Sumber: Kurniawan, 2020)

Figure 5 and Figure 6 show the loss graph and accuracy graph for the training process and the validation process. The training process on the loss chart shows the initial results of the loss above 10% at the beginning of the training process then decreases by less than 10 iterations and the results of the loss for the validation process show the results of 5% at the beginning of the training process. The accuracy results during the training process showed a result of 97% at the beginning of the training process then increased to 99% after going through 10 iterations.

3.2 CNN Accuracy Testing Against Training Process Iteration Variations

The purpose of this test is to see the effect of CNN's accuracy in detecting DoS attacks on variations in the length of the training process with varying data. The variation in the training process is meant to change the number of iterations. The results of tests carried out using 5 different data from the previous test. The training process is carried out 10 times for each data with the number of iterations between 10 and 100 with multiples of 10.

Table 3.The Results of The 1st Test for CNN's Accuracy Against Variations In The Training Process Iteration

No	Total Data	Long Training Process (Total Iterasi)	Network Activity		Long Time (second)	Loss (%)	Accuracy (%)
			Normal (%)	DoS (%)			
1	35738	10	61.05 %	38.95 %	82	5.76 %	99.03 %
2		20	61.32 %	36.68 %	161	5.29 %	98.87 %
3		30	62.59 %	37.41 %	241	4.72 %	98.78 %
4		40	61.64 %	38.36 %	326	4.68 %	98.92 %
5		50	60.62 %	39.38 %	400	4.57 %	98.94 %
6		60	61.82 %	38.18 %	484	4.55 %	98.99 %
7		70	62.60 %	37.40 %	569	5.39 %	98.92%
8		80	60.41 %	39.59 %	661	6.99 %	98.99 %
9		90	61.40 %	38.60 %	789	7.11 %	98.85 %
10		100	62.19 %	37.81 %	1056	5.28 %	98.92 %

Source : Kurniawan, 2020)

Table 4. The Results of The 2nd Test for CNN's Accuracy Against Variations In The Training Process Iteration

No	Total Data	Long Training Process (Total Iterasi)	Network Activity		Long Time (second)	Loss (%)	Accuracy (%)
			Normal (%)	DoS (%)			
1	70352	10	55.28 %	44.72 %	211	3.77 %	99.41 %
2		20	55.77 %	44.23 %	322	3.17 %	99.32 %
3		30	55.43 %	44.57 %	323	3.04 %	99.40 %
4		40	56.63 %	43.37 %	631	3.17 %	99.45 %
5		50	56.02 %	43.98 %	793	2.11 %	99.52 %
6		60	56.13 %	43.87 %	1197	3.56 %	99.37 %
7		70	55.67 %	44.33 %	1440	6.12 %	99.35%
8		80	55.99 %	44.01 %	1636	4.60 %	99.45 %
9		90	55.99 %	44.01 %	2202	3.60 %	99.32 %

10		100	55.59 %	44.41 %	1919	3.60 %	99.55 %

Source : Kurniawan, 2020)

Table 5. The Results of The 3rd Test for CNN's Accuracy Against Variations In The Training Process Iteration

No	Total Data	Long Training Process (Total Iterasi)	Network Activity		Long Time (second)	Loss (%)	Accuracy (%)
			Normal (%)	DoS (%)			
1	47285	10	76.17 %	22.83 %	136	4.51 %	99.18 %
2		20	75.52 %	24.48 %	214	4.57 %	98.97 %
3		30	75.14 %	24.86 %	320	3.87 %	99.35 %
4		40	75.59 %	24.41 %	426	3.70 %	99.24 %
5		50	75.65 %	24.35 %	576	3.47 %	99.23 %
6		60	74.83 %	25.17 %	755	3.14 %	99.41 %
7		70	75.66 %	24.34 %	810	2.99 %	99.38%
8		80	75.18 %	28.82 %	859	4.71 %	99.28%
9		90	75.40 %	24.60 %	963	4.36 %	99.28%
10		100	75.33 %	24.67 %	12467	5.82 %	97.96 %

Source : Kurniawan, 2020)

Table 6. The Results of The 4th Test for CNN's Accuracy Against Variations In The Training Process Iteration

No	Total Data	Long Training Process (Total Iterasi)	Network Activity		Long Time (second)	Loss (%)	Accuracy (%)
			Normal (%)	DoS (%)			
1	44042	10	57.59 %	42.41 %	117	4.02 %	98.97 %
2		20	57.17 %	42.83 %	256	4.23 %	99.13 %
3		30	57.11 %	42.89 %	380	4.94 %	98.94 %
4		40	57.96 %	42.04 %	472	3.91 %	99.04 %
5		50	56.71 %	43.29 %	498	4.59 %	98.99 %
6		60	57.46 %	42.54 %	715	4.82 %	99.18 %
7		70	57.69 %	42.31 %	707	6.49 %	99.14%
8		80	58.60 %	41.40 %	1035	3.63 %	99.10%
9		90	57.42 %	42.58 %	910	4.21 %	99.02%
10		100	57.49 %	42.51 %	1157	5.52 %	98.98%

Source : Kurniawan, 2020)

Table 7. The Results of The 5th Test for CNN's Accuracy Against Variations In The Training Process Iteration

No	Total Data	Long Training Process (Total Iterasi)	Network Activity		Long Time (second)	Loss (%)	Accuracy (%)
			Normal (%)	DoS (%)			
1	51921	10	70.49 %	29.51 %	150	4.87 %	99.27 %
2		20	70.39 %	29.61 %	275	4.31 %	99.36 %
3		30	71.11 %	28.89 %	436	4.02 %	99.40 %
4		40	70.07 %	29.93 %	530	4.74 %	99.43 %
5		50	70.73 %	29.27 %	656	3.47 %	99.51 %
6		60	69.51 %	30.49 %	791	4.35 %	99.33 %
7		70	70.40 %	29.60 %	800	4.85 %	99.55%
8		80	70.47 %	29.53 %	1022	3.64 %	99.37%
9		90	70.58 %	29.42 %	1045	4.61 %	99.44%

P-ISSN: 2528-0260 E-ISSN: 2579-5392

10		100	70.59 %	29.41 %	1350	5.78 %	98.83%

Source : Kurniawan, 2020)

Table 3 to table 7 shows the results of the CNN test on the variation iteration of the training process. From table 3 to table 7, it can be seen that the percentage results of network activity have begun to vary with the variation in the input data entered into CNN. The length of time the training process takes the longer the number of iterations increases. However, the amount of data entered can affect the length of time the training process takes. For the least amount of data in the first test, the time required for the training process with 100 iterations is 1056 seconds or 17 minutes 36 seconds. Whereas in the second test with the most amount of data it took 1919 seconds or 31 minutes 59 seconds.

For the resulting accuracy results an average of 99.17% and an average of 4.46%. In several experiments the amount of data affects the results of the CNN process carried out, by having a large number of datasets the better the results will be obtained.

3.3 TPR and FPR Testing of DoS Attack Detection Results

The purpose of this test is to know and see the results of the ROC Curve. This ROC Curve is a graph showing the result of the value between the True Positive Rate and the False Positive Rate. This ROC Curve value will be used to find out how well the CNN system is correct in predicting.

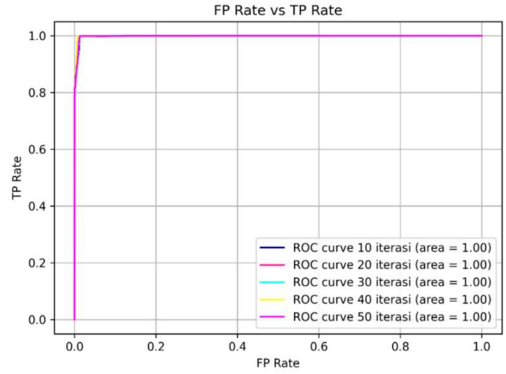


Figure 7. The ROC Curve Graphic
(Source: Kurniawan, 2020)

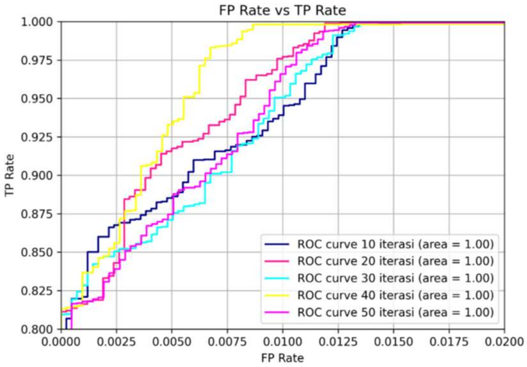
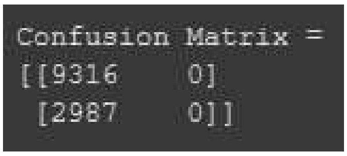


Figure 8. The ROC Curve Graph Is Enlarged at The Top Left
(Source: Kurniawan, 2020)

Figure 7 and Figure 8 show the results of the TPR and FPR testing of the DoS attack detection results. From the test results, each test data was carried out for 5 experiments with different iterations with the aim of whether the effect of different iterations could provide significant results on TPR and FPR. Figure 7 is the original graph when plotted and Figure 8 is the result of the graph that has been enlarged to the upper left to see in more detail, from Figure 7 the graph results can be seen that the ROC Curve value is obtained at 1.For the graph results from Figure 7, this coverage is visible because CNN requires more datasets to input and CNN only detects 2 classes whether it is a DoS attack or not a DoS attack.



```
Confusion Matrix =
[[9316    0]
 [2987    0]]
```

Figure 9. Confusion Matrix
(Source: Kurniawan, 2020)

Confusion matrix results are obtained where the confusion matrix output is in the form of a table in which the column section is the true and false prediction column, and in the row section is the actual true and false row. The confusion matrix value in each of these test results is related to the ROC Curve graph. By using the values from the confusion matrix, values for other parameters such as accuracy, precision, TPR, FPR can be generated.

CONCLUSION

The conclusions obtained from the research results, IDS using deep learning for detection of DoS attacks are as follows :

1. By using wireshark log data and through data processing, the normalization of IDS implementation using deep learning can be done.
2. IDS results using Deep Learning can detect network activity between Dos attacks and non-DoS attacks in the form of a bar graph.
3. In testing the CNN accuracy of data variations by running 5 test data, the results obtained an average accuracy of 99.32% with an average loss of 4.08%.
4. In the CNN accracy test of the various iterations of the training process, 5 test data were run with each data being carried out 10 experiments with different iterations, then the results were obtained that the increasing number of iterations the time it takes to conduct training will be longer and the amount of data inputted will also affect the length of time the training process. For the resulting accuracy an average of 99.17% with an average loss of 4.46%.
5. The results of the True Positive Rate and False Positive Rate are obtained by running 5 test data where each data is carried out 5 times with different iterations then the results of 5 experiments are presented in one graph, with the area value under the ROC Curve for the 5 test data of 1.

SUGESSTION

For further research on IDS using deep learning, the suggestions to be conveyed are as follows :

1. Added the attack method used so that it can test CNN in detecting variations in the types of attacks that are added.
2. The inputted data can be added even more, by adding attack methods and the data inputted more, can be further analyzed for the results of loss, accuracy, FPR, and TPR.

3. For maximum results, the implementation of IDS using Deep Learning can be done on a real computer network so that the data that will be inputted into CNN corresponds to real data that occurs in actual computer network activities.

REFERENCES

[1] H. Corresponding, “INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH A REVIEW OF MACHINE LEARNING TECHNIQUES EFFICIENCY IN DOS ATTACK Zerina Masetic * Nejdet Dogru,” no. 12, 2017.

[2] A. A. Kurniawan, “Intrusion Detection System Menggunakan Deep Learning Untuk Deteksi Serangan DoS,” *Intrusion Detect. Syst. Menggunakan Deep Learn. Untuk Deteksi Serangan DoS*, pp. 39–57, 2020.

[3] U. Michelucci, *Advanced applied deep learning: Convolutional neural networks and object detection*. 2019.

[4] T. Ahmad, M. A. Anwar, and M. Haque, “Machine Learning Techniques for Intrusion Detection,” pp. 47–65, 2020, doi: 10.4018/978-1-7998-2242-4.ch003.

[5] Abdi, A. P, (2019), *Bobol Akun e-Banking, Pemilik Kehilangan Uang Rp1 Miliar* (Online), From https://tirto.id/bobol-akun-e-banking-pemilik-kehilangan-uang-rp1-miliar-efZZ

[6] Franedya, R, (2019), *Hati-hati Pakai WiFi Gratisan, HP Bisa Diretas Hacker Jahat*. (Online), From https://www.cnbcindonesia.com/tech/20190830060632-37-95838/hati-hati-pakai-wifi-gratisan-hp-bisa-diretas-hacker-jahat

[7] Sena, S. (2017, November 13), *Introduction Deep Learning Part 7 : Convolutional Neural Network (CNN)*. (Online), From https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94

[8] Sinh-Ngoc Nguyen, Van-Quyet Nguyen, Jintae Choi, and Kyungbaek Kim. 2018. Design and implementation of intrusion detection system using convolutional neural network for DoS detection. In <i>Proceedings of the 2nd International Conference on Machine Learning and Soft Computing</i> (<i>ICMLSC '18</i>). Association for Computing Machinery, New York, NY, USA, 34–38. DOI:https://doi.org/10.1145/3184066.3184089