DDoS Detection Using Artificial Neural Network Regarding Variation of Training Function

Article *in* Journal of Computational and Theoretical Nanoscience · December 2018

DOI: 10.1166/asl.2018.12117

3 authors:

Imam Riadi
Ahmad Dahlan University
105 PUBLICATIONS **389** CITATIONS
SEE PROFILE

Sunardi Sunardi
Ahmad Dahlan University
46 PUBLICATIONS **42** CITATIONS
SEE PROFILE

Arif Wirawan Muhammad
10 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Advanced Science Letters
Vol. 4, 3398–3402, 2011

DDoS Detection Using Artificial Neural Network Regarding Variation of Training Function

Imam Riadi[1], Sunardi[2], Arif Wirawan Muhammad[3]

[1]*Department of Information System, Ahmad Dahlan University, Yogyakarta, Indonesia*
[2]*Department of Electrical Engineering, Ahmad Dahlan University, Yogyakarta, Indonesia*
[3]*Department of Information Technology, Ahmad Dahlan University, Yogyakarta, Indonesia*

Distributed denial-of-service (DDoS) is a type of network attack with the number both in volume and intensity has increased significantly in recent years. DDoS is a major problem for the integrity, secrecy and availability of resources owned by Internet organizations. Early detection of DDoS attacks is a fundamental process performed automatically by the Intrusion Detection System (IDS), which generally uses signature-based detection techniques that can be said to be far from perfect when compared with the more modern cyber attack techniques. This study proposed DDoS detection system based on network feature which produced from statistical extraction combined with artificial neural network (ANN) method as detection engine with training function variation. The experiment resulted that Quasi-Newton training function (Matlab trainlm) give the highest accuracy value 0.992 (99.2%) against Resilient-Propagation training function (Matlab trainrp) which resulted accuracy at 0.989 (98,9%) and the Scaled-Conjugate training function (Matlab trainscg) which resulted accuracy at 0.988 (98,8%).

Keywords: DDoS, Neural Network, Training Function, Statistical Extraction

Submitted: 20 June 2017. Accepted: 17 July 2017. Revised: 17 July 2017

1. INTRODUCTION

Distributed denial-of-service (DDoS) is a type of network attack with the number both in volume and intensity has increased significantly in recent years[1]. DDoS was reported as the most frequent attack in 2016. The number of DDoS attacks increased from 35% in 2015 to 51% by 2016[2]. DDoS is a major problem for the integrity, secrecy and availability of resources owned by Internet organizations[3].

Early detection of DDoS attacks is a fundamental process performed automatically by the Intrusion Detection System (IDS), which generally uses signature-based detection techniques that can be said to be far from perfect when compared with the more modern cyber attack techniques. The IDS detection system generally only monitors and tags the suspicious activity of the network and immediately reported as an alert, resulting in the impact of a large volume of alerts with a high false negative or high false positive rate and low accuracy due to network traffic is a non-stationary data[4]. The scale and cost to tackle DDoS attacks in cyberspace almost doubled from the previous year. Studies from ArborNetwork and Akamai reinforce the notion that stopping DDoS is impossible[5]. The use of statistical and neural network methods to detect DDoS attacks in a cloud environment has been implemented by[6]. In the study[6], packet attribute *TTL, total length, source port, destination port, protocol type, window size,* dan *flag* resulted from the sniffing capture process in a given time window. Those attibutes extracted with the Jensen-Shannon Divergence statistical concept that generates the value of information divergence. The supervised learning decision-tree and naïve bayes methods are also used to detect backscatter data flow from DDoS attacks based on the CAIDA 2008 dataset[7] and conclude that each decision-tree and naïve bayes method produces a fairly high recognition performance even though it does not use the detection base feature such as IP address and port number[7]. The study[8] use a supervised learning neural network method to detect DDoS attacks in Hadoop and HBase environments. Detection features average CPU usage in the server, the average packet size, and the total number of TCP connections, were used in[8] and resulted in a conclusion that the supervised learning neural network method successfully established appropriate alert according to training group.

However, in real implementation, the detection feature used is too weak, because one character of DDoS attack is burst and non-stationary. It is therefore necessary to make improvements in the detection feature[8].

Based on earlier research this study proposed DDoS detection system based on network feature which produced from statistical extraction combined with artificial neural network (ANN) method as detection engine. The variation of training function aiming to find out the suitable training function for DDoS detection. This study use network DDoS dataset published by the Center for Applied Internet Data Analysis (CAIDA) and network DDoS dataset published by Ahmad Dahlan University Networks Laboratory[9]. CAIDA dataset and Ahmad Dahlan University Networks Laboratory dataset form in a .pcap file format, generated from packet sniffer software installed on network routers with star topology.

2. EXPERIMENTAL DETAILS

The study of DDoS detection using artificial neural network (ANN), involves steps as seen on Fig. 1.

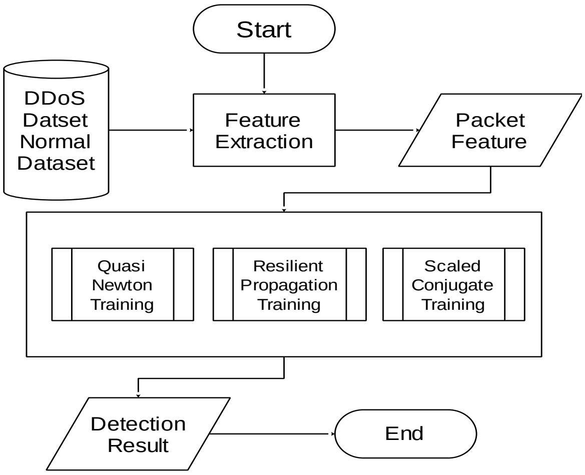
Start
DDoS Datset Normal Dataset
Feature Extraction
Packet Feature
Quasi Newton Training
Resilient Propagation Training
Scaled Conjugate Training
Detection Result
End

Fig. 1. Steps of DDoS Detection

- Get network DDoS dataset from CAIDA and Ahmad Dahlan University Networks Laboratory in .pcap format.
- Extract network packet feature, using statistical method.
- Training ANN with Quasi-Newton, Resilllent-Propagation and Scaled-Conjugate function.
- Comparison on classification result using parameters accuracy, mean-squared error (mse), and iteration.

To detect DDoS attack, first network feature must be extracted from dataset. The aim of feature extraction is to measure certain attributes in original data that distinguish one input pattern from another pattern[10]. This study extracted network packet to six features based on statistical method. The six features are :

- Average packet size.

  The longer DDoS attack occurs, always followed by a rise in the value of average packet size[11].
- Number of packets

  DDoS attacks overwhelm target by sending many packets at a certain time lag. DDoS resulting high number of the packet[11].
- Time interval variance

  DDoS attack send large numbers of packages in a certain time span. Thus affected the value of time interval variance will be smaller and nearly zero. Time interval variance stated as Equation (1)[12].

  $$t_c{}^2 = \frac{\sum(t_n - t)^2}{n} \qquad (1)$$

  Where t_n is time of a packet received and $\bar{t}$ is the rate of time a packet is received.
- Packet size variance

  The normal traffic resulted high value of packet size variance. In other hands, DDoS attacks resulted close to zero packet size variance value, caused by the monotony of packet size that sent to target. Packet size variance stated as Equation (2)[12].

  $$p_c = \sqrt{\frac{\sum(p_n - \bar{p})^{\,2}}{n}} \qquad (2)$$

  Where p_n is received packet size, and $\bar{p}$ is packet size rate.
- Packet rate.

  Packet rate stated as the number of packets sent by the source computer to a destination computer within a specific time frame as seen on Equation (3)[12].

  $$p_c = n_p \; x \; \frac{1}{(t_e - t_x)} \qquad (3)$$

  Where n_p is the number of packets, t_e is end time a packet is received, t_s is the initial time a packet is received.
- Number of bytes.

  DDoS attack always increases the number of bytes in constant[12].

There are many numbers of batch training algorithms which can be used in Artificial Neural Network[13]. Algorithm that most used for training are :

- Newtonian training function[14 15] and the derivatives that called a Quasi-Newton method (Matlab trainlm). [16 17].
- Resilllent-Propagation training function (Matlab trainrp) refers to the gradient-descent algorithm [15].
- Scaled-Conjugate training function (Matlab trainscg) refers to the conjugate-gradient algorithm that exploits the gradient's negative to achieve convergence [15] .

There is no certainty to determine the best number of neurons and hidden layers used to resolve a problem with a ANN[18]. Based on that reason, this study use Kolmogorov theory that stated the number of neuron in hidden layer must be 2n+1 , where n is the number of input neuron[11]. Accuracy, mean-squared error, and iteration parameters was used in this research for classification performance analysis.

- Accuracy is the number of ratio between DDoS and normal packet data recognition result compared to the overall packet data.

- Mean-squared error (mse), is an absolute error of ANN actual output pattern compared with desired output pattern.
- Iteration is the time value takes by ANN to reach its convergence [16].

The ANN output layer in this study configured as one node. The output will be 0 stated as normal condition, or 1 stated as DDoS condition.

4. RESULTS AND DISCUSSION

The experiments carried out on Matlab 2013b environment running on Windows 10 64-bit. The dataset consists of 200 traffic data of DDoS condition and 200 traffic data of normal condition with six network features. Dataset for ANN training phase was divided into 70% sets for training, 15% sets for validation, and 15% sets for testing. The sample pattern distribution of dataset for training, validation, and testing was created by Matlab dividerand random function to avoid tendency of bias. Tangen sigmoid transfer function used in the hidden layer and pure linear transfer function used in output layer. The basic parameters epoch = 10000, performance function = mse, goal = 0.01, maximum fail = 5, minimum gradient = 1.00e-10, mu = 1.00e+10 were used in the training process. Here the result for each training function. Quasi-Newton method (Matlab trainlm) training result presented on Fig.2

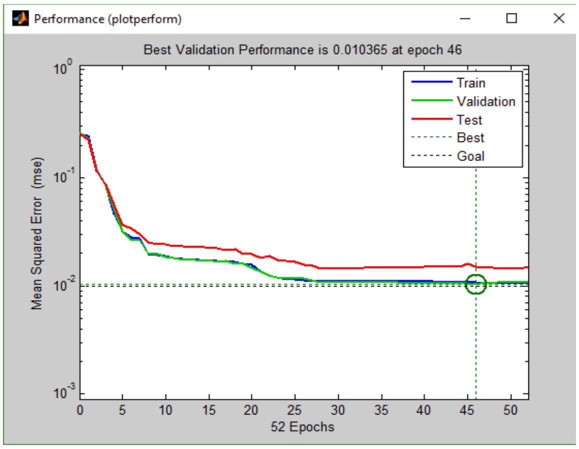

Fig. 2. Quasi-Newton Training Result

Scaled-Conjugate method (Matlab trainscg) training result for ANN layer 6-(3)-2 presented on Fig.3.

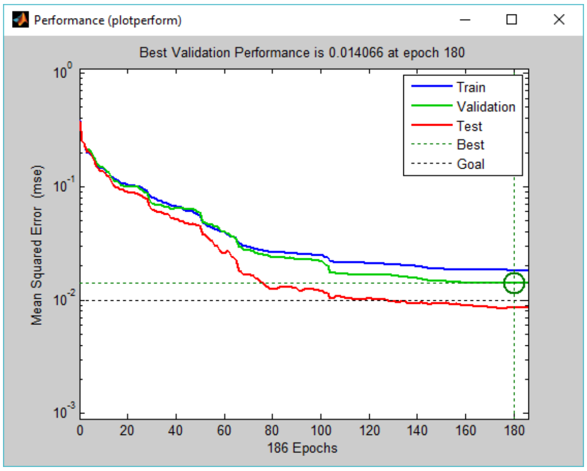

Fig. 3. Scaled-Conjugate Training Result

Resilient-Propagation method (Matlab trainrp) training result for presented on Fig.4.

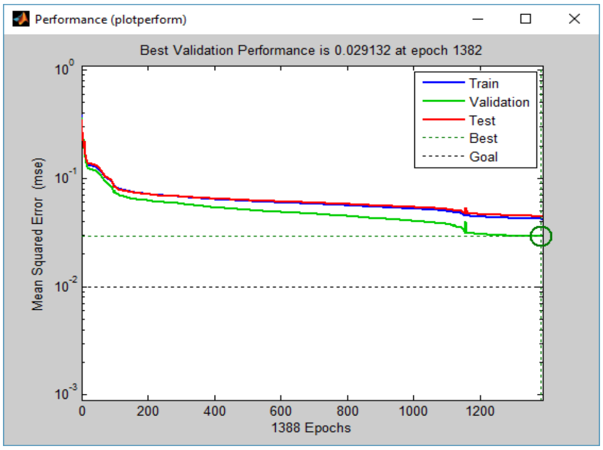

Fig. 4. Resilient-Propagation Training Result

All training result stated that there was no overtraining faced on ANN scheme. Quasi-Newton training function (Matlab trainlm) resulted the highest accuracy value 0.992 (99.2%) on CAIDA dataset and 0.984 (98.4%) on Ahmad Dahlan University Networks Laboratory dataset against other training function as stated on Fig.5 and Fig.6. ANN with Resilient-Propagation method (Matlab trainrp) resulted accuracy at 0.989 (98.9%) on CAIDA dataset and 0.979 (97.9%) on Ahmad Dahlan University Networks Laboratory dataset. However, the Scaled-Conjugate training function (Matlab trainscg) resulted less accuracy value at 0.988 (98.8%) on CAIDA dataset and 0.972 (97.2%) on Ahmad Dahlan University Networks Laboratory dataset. Based on Fig.5 and Fig.6 Kolmogorov's theory that stated the best number of hidden layer neurons to solve ANN problem is 2n + 1 where n is the number of of input nurons produce high classification accuracy on CAIDA dataset and Ahmad Dahlan University Networks Laboratory dataset.

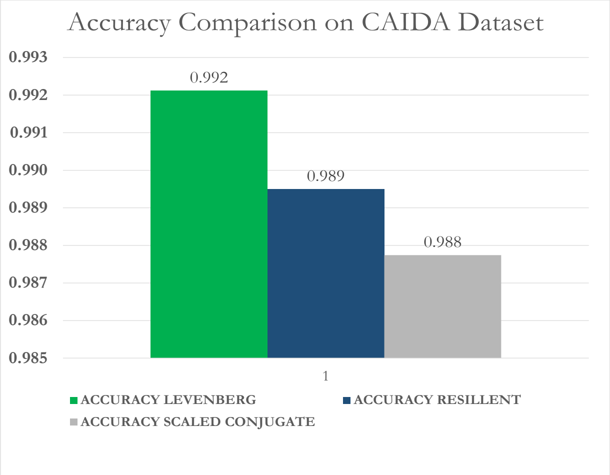

Fig. 5. Accuracy Comparison on CAIDA Dataset

From Fig.7 and Fig.8 we can conclude that Quasi-Newton (Matlab trainlm) training function resulted small average mse value on ANN compared to the Scaled-Conjugate (Matlab trainscg) and Resilient-Propagation (Matlab trainrp) training functions on CAIDA Dataset and also on Ahmad Dahlan University Laboratory dataset.

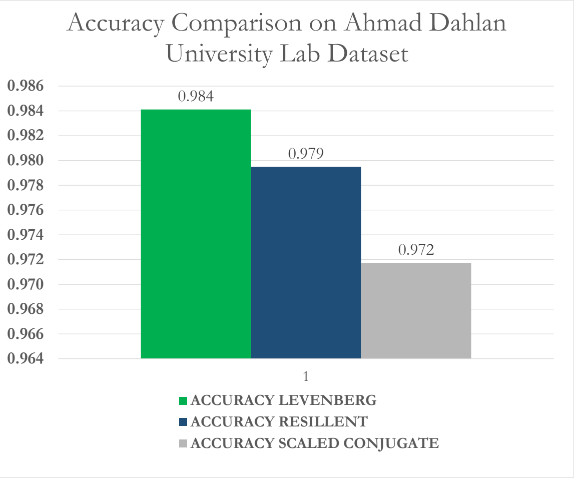

Fig. 6. Accuracy Comparison on Ahmad Dahlan University Lab Dataset

ANN with Quasi-Newton (Matlab trainlm) training function has fewer iterations compared to Scaled-Conjugate (Matlab trainscg) and Resilient-Propagation (Matlab trainrp) training functions for all ANN schemes.

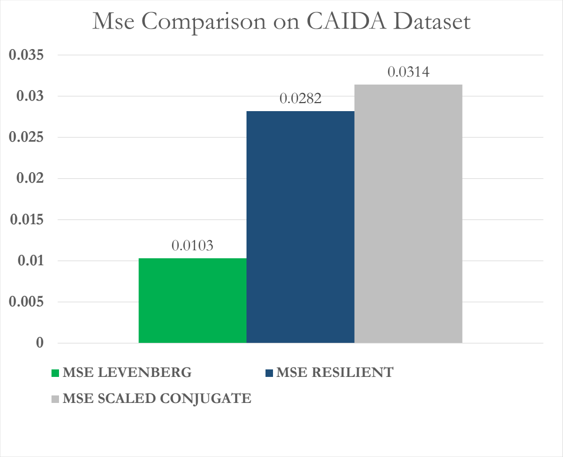

Fig. 7. Mse Comparison on CAIDA Dataset

Based from Fig.9 and Fig.10 we can conclude that ANN with Quasi-Newton (Matlab trainlm) training function is fast to reach convergence and efficient in time.

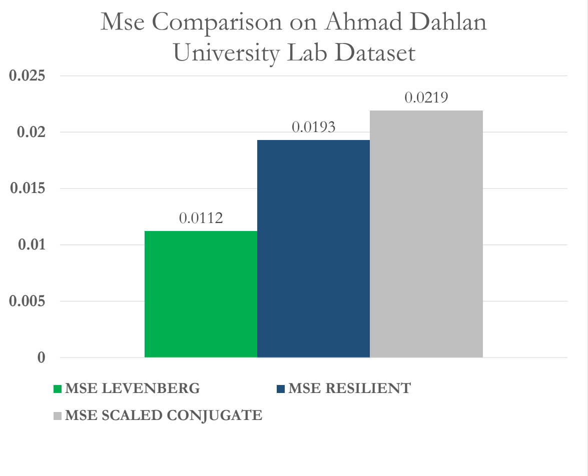

Fig. 8. Mse Comparison on Ahmad Dahlan University Lab Dataset

ANN with Scaled-Conjugate (Matlab trainscg) and Resilient-Propagation (Matlab trainlm) training function achieve convergence in a slower time compared to Quasi-Newton (Matlab trainlm) training function on CAIDA dataset and Ahmad Dahlan University Laboratory dataset as stated in Fig.9 and Fig.10

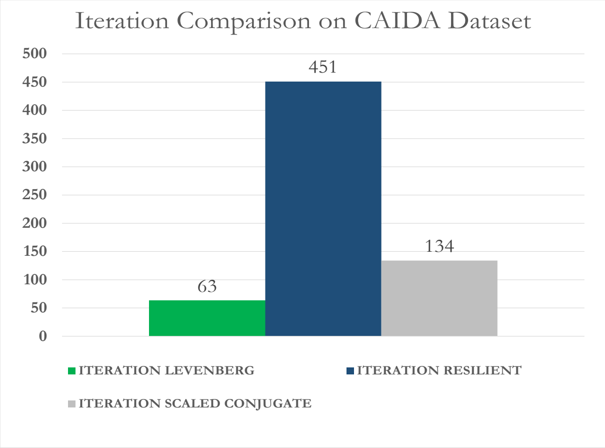

Fig. 9. Iteration Comparison on Ahmad Dahlan University Lab Dataset

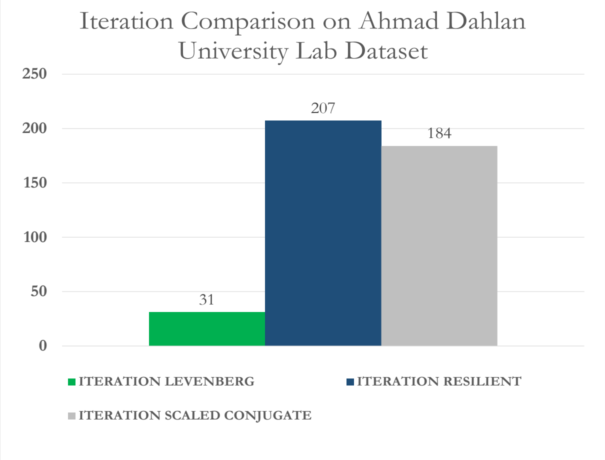

Fig. 10. Iteration Comparison on Ahmad Dahlan University Lab Dataset

4. CONCLUSIONS

DDoS detection can be done effectively with artificial neural network (ANN) with the appropriate training functions. This study found best DDoS detection accuracy was 99.2% given by ANN by the number of hidden layer neurons 2n+1 where n is the number of input nurons with Quasi-Newton (Matlab trainlm) training function. This study found significant differences in accuracy percentage by applying training function. Quasi-Newton (Matlab trainlm) training resulted best accuracy among other training function (Scaled-Conjugate and Resilient-Propagation training function) and fast to reach convergence. Further improved study must be done on other ANN parameters like increasing size of input, variating the numbers of hidden layes, reducing goal mse, and use other training method.

Acknowledgments: This research is supported by the Graduate Program of Ahmad Dahlan University Indonesia.

References and Notes

1 I. Riadi, A. W. Muhammad, and Sunardi, “Integrasi Metode Normalized Relative Network Entropy dan Neural Network Backpropagation (BP) untuk Deteksi dan Peramalan Serangan DDoS,” National. 4th APPPTM Indonesian Conference, vol. 4, (2016).

2 A. Networks, “World Wide Infrastrucure Security Report 2015,” (2016).

3 A. Saied, R. E. Overill, and T. Radzik, “Detection of Known and Unknown DDoS Attacks using Artificial Neural Networks,” Neurocomputing, vol. 172, pp. 385–393, (2015).

4 H. S. Kim and H. K. Kim, “Network Forensic Evidence Acquisition (NFEA) With Packet Marking,” Ninth IEEE International Symposium Parallel Distribution Processes with Application Working Network, pp. 389–394, (2011).

5 Arbor Networks, “Worldwide Infrastructure Security Report,” vol. IX, pp. 1–83, (2014).

6 M. Zareapoor, “Statistical-Based Filtering System Against DDOS Attacks in Cloud Computing,” IEEE International Conference Advanced Computing Communicaton Informatics, pp. 1234–1239, (2014).

7 E. Balkanli, J. Alves, and A. N. Zincir-Heywood, “Supervised Learning to Detect DDoS Attacks,” IEEE International Conference of Advenced Computing Communication and Informatics, (2014).

8 T. Zhao, D. C. T. Lo, and K. Qian, “A Neural Network Based DDoS Detection System Using Hadoop and HBase,” Proc. - 2015 IEEE 17th International. Conference on High Performance Computing and Communication, pp. 1326–1331, (2015).

9 I. Riadi, A. W. Muhammad, and Sunardi, “Neural Network-Based DDoS Detection Regarding Hidden Layer Variation,” Journal of Theoretical and Applied Information Technoogy, vol. 95, pp. 1–9, (2017).

10 I. Riadi, A. W. Muhammad, and Sunardi, “Network Packet Classification Using Neural Network Based on Training Function and Hidden Layer Neuron Number Variation,” International Journal of Advanced Computer Science and Application, vol. 8, no. 6, pp. 1–4, (2017).

11 C. J. Hsieh and T. Y. Chan, “Detection DDoS Attacks Based On Neural-Network Using Apache Spark,” 2016 International Conference on Applied System Innovation IEEE ICASI 2016, pp. 1–4, (2016).

12 T. P. Thwe Thwe Oo, “A statistical approach to classify and identify DDoS attacks using UCLA dataset,” International Journal of Advanced Computer Engineering Technology, vol. 2, no. 5, p. 1766, (2013).

13 N. Pise and P. Kulkarni, “Algorithm Selection for Classification Problems,” SAI Computing Conference 2016, pp. 203–211, (2016).

14 Y. H. Hu and J.-N. Hwang, Handbook of Neural Network Signal Processing, First Edit. New York: CRC Press, (2002).

15 S. Haykin, Neural Networks and Learning Machines, vol. 3. (2008).

16 M. Anthony and P. L. Bartlett, Neural Network Learning : Theoretical Foundations, First Edit. New York: Cambridge University Press, (2009).

17 F. Soares and A. M. F. Souza, Neural Network Programming With Java : Unleash The Power Of Neural Networks By Implementing Professional Java Code. (2016).

18 C.-J. Hsieh and T.-Y. Chan, “Detection DDoS Attacks Based on Neutral-Network Using Apache Spark,” National Chin-Yi University of Technology. Taichung, Taiwan, pp. 1–4, (2015).