International Journal Of Artificial Intelegence Research
Vol 4, No 1, June 2020, pp. 01 - 08
ISSN: 2579-7298

Machine Learning-Based Distributed Denial of Service Attack Detection on Intrusion Detection System Regarding to Feature Selection

Arif Wirawan Muhammad [a,1,*], Cik Feresa Mohd Foozy [b,2], Ahmad Azhari [c,3]

[a] Institut Teknologi Telkom Purwokerto, Jl DI Pandjaitan 128 Karangreja, Banyumas 53147, Indonesia
[b] Universiti Tun Hussein Onn, Delta Road 1/6 Batu Pahat, Johor 86400, Malaysia
[c] Universitas Ahmad Dahlan, Jl Ringroad Selatan Kragilan, Yogyakarta 55191, Indonesia
[1] arif@ittelkom-pwt.ac.id*; [2] feresa@uthm.edu.my; [3] ahmad.azhari@tif.uad.ac.id
* corresponding author

ARTICLE INFO

Article history:
Received *December 2019*
Revised *Feb 2020*
Accepted *March 2020*

Keywords:
IDS
DDoS
Feature
Machine Learning
Classification

ABSTRACT

Distributed Service Denial (DDoS) is a type of network attack, which each year increases in volume and intensity. DDoS attacks also form part of the major types of cyber security threats so far. Early detection plays a key role in avoiding the catastrophic effects on server infrastructure from DDoS attacks. Detection techniques in the traditional Intrusion Detection System (IDS) are far from perfect compared to a number of modern techniques and tools used by attackers, because the traditional IDS only uses signature-based detection or anomaly-based detection models and causes a lot of false positive flags, since the flow of computer network data packets has complex properties in terms of both size and source. Based on the deficiency in the ordinary IDS, this study aims to detect DDoS attacks by using machine learning techniques to enhance IDS policy development. According to the experiment the selection of features plays an important role in the precision of the detection results and in the performance of machine learning in classification problems. The combination of seven key selected dataset features used as an input neural network classifier in this study provides the highest accuracy value at 97.76%.

1. Introduction

Distributed denial of service (DDoS) is a type of network attack that continues to increase every year, in terms of volume and intensity [1]. DDoS attacks pose a threat to Internet users and all the infrastructure that is in them, including bandwidth, server resources, data integrity, data availability, and confidentiality of data stored on the server [2]. Until now DDoS attacks are still included in the main types of cyber security threats. Early detection plays a fundamental role in preventing the fatal impact of DDoS attacks on server resources. One of the basic actions taken to prevent DDoS attacks is to install an Intrusion Detection System (IDS) on the server to monitor the flow of data packets that enter the internal network or vice versa [3]. Detection techniques in common IDS are far from perfect when compared to a variety of modern techniques and tools used by attackers because IDS still uses signature-based detection or anomaly-based detection models [4]. The use of detection models in both signature-based and anomaly-based IDs has a high false-positive rate. From a technical point of view, signature-based IDs and anomaly-based IDS work by monitoring the flow of data packets that enter or exit the internal network. IDS will provide a marker if it finds data flow activities that do not match the signature database that has been embedded in the IDS [5]. Thus detection model logically will cause a lot of false positive flags, because the flow of computer network data packets has dynamic properties both in terms of size, source, protocol, and content of data contents [6]. On the other hand, signature-based IDS and anomaly-based IDS have two main weaknesses. The first weakness is when

IDS detects attacks that begin with the SYN protocol, for example SYN-Flood, because the SYN protocol is a legal and absolute protocol to be used to initiate communication between two computers/devices in a network [7]. Therefore, ordinary IDS is difficult to generate alerts against attacks that begin with the SYN protocol artifact. The second weakness of IDS is mainly due to the TCP/IP protocol deficit which makes it easy for an attacker to start a DDoS attack for example by using the Ping command which is available by default throughout the operating system or using special tools such as HOIC, LOIC, XOIC, golden-eye, and etc [8]. The use of TCP/IP standard protocols by attackers to carry out DDoS attacks causes the target too slow to realize that it is under attack, so that it also impacts process of attack mitigation [9]. Weaknesses of the TCP/IP protocol are difficult to handle by ordinary IDS. In addition, the high volume of false positive flags generated by ordinary IDS has quite an impact on server hardening efficiency. Based on the weaknesses that exist in the ordinary IDS, this study aims to detect DDoS attacks by utilizing machine learning techniques so that it can be an improvement in the development of IDS devices. This research utilizes a DDoS attack dataset sourced from UNSW-NB15 (University of New South Wales) [10] for further processing by applying the neural network method to produce DDoS detection machine learning models.

2. Detection Approach

The DDoS attack detection approach implemented in this study is divided into several stages namely :

2.1 Retrieving Dataset

The first step is getting the UNSW-NB15 DDoS attack dataset published by the University of New South Wales. The UNSW-NB15 dataset is the latest attack dataset containing the attack packet flow record and a normal packet in the form of a tcpdump file, recording the data flow for 31 hours [11]. The attack packet flow is synthetically simulated using IXIA software, mimicking attacks with high-speed low footprinting. There are nine types of attacks covered by the UNSW-NB15 dataset, presented in Table 1. The grouping of UNSW-NB15 dataset feature categories is carried out systematically, namely flow features, basic features, data packet content features, time features, and additional features. Basically, the motivation for the formation of the UNSW-NB15 dataset is to improve the issue of shortcomings in the KDDCUP99 and NSLKDD datasets. [12].

Table.1 UNSW-NB15 Flow Record

No.	Flow Record Type	Remarks
1.	DDoS	Attempts to degrade server resources cannot be accessed by authorized users. Interruption of services provided by the server to the host.
2.	Analysis	The act of searching for a server or host system weak point on the network. For example scan, spam, html injection business.
3.	Fuzzers	Actions that cause network communication or running programs to be delayed temporarily, by injecting random data.
4.	Backdoors	The technique of bypassing a system security door secretly to access a machine and the data it contains.
5.	Generic	The technique of disrupting encrypted data flow.
6.	Exploit	Attempts to exploit network or software security holes on the server or host.
7.	Reconnaissance	The process of snooping on security holes on a network or server by gathering information related to an attack.
8.	Worms	Attempts to replicate malicious code or software that an attacker has implanted into the infected network or machine. Replication aims to spread malicious code or software to other machines that haven't been infected.
9.	Shellcode	A small piece of malicious code that is used as a carrier of information / triggers of an attack / exploitation.
10.	Normal	Natural transaction data flow.

2.2 Selecting Feature

In this study, the type of record that will be analyzed will be specific to the DDoS record group as presented in Table 1. The UNSW-NB15 DDoS dataset attack record has features as presented in Table 2.

Table.2 UNSW-NB15 DDoS Features

Features	DDoS Features		
	Feature No.	***Type***	***Description***
Srcip	1	nominal	Source of IP address
Sport	2	integer	Source of port number
Dstip	3	nominal	Destination of IP address
dsport	4	integer	Destination of port number
Proto	5	nominal	Transaction protocol
State	6	nominal	Indicates to the state and its dependent protocol
Dur	7	Float	Record total duration
sbytes	8	Integer	Source to destination transaction bytes
dbytes	9	Integer	Destination to source transaction bytes
Sttl	10	Integer	Source to destination time to live value
Dttl	11	Integer	Destination to source time to live value
Sloss	12	Integer	Source packets retransmitted or dropped
Dloss	13	Integer	Destination packets retransmitted or dropped
Service	14	nominal	http, ftp, smtp
Packet_Label	15	binary	0 for normal and 1 for attack records

The fifteen features are then selected using the Information Gain technique with the aim of reducing computational time and obtaining a high-accuracy machine learning model. Information Gain is the amount of mutual information obtained from a combination of observational variables and is a divergence from the Kullback-Liebler theory [13]. In terms of machine learning, Information Gain is useful for selecting and selecting several important features based on theories that measure the value of information possessed by a feature related to other features. For an "a" feature, information gain is the amount of entropy contained by "a" compared to the "c" feature of all available features [14]. Important features are indicated by the maximum value of entropy possessed by the feature. The Information Gain equation is presented in (1).

$$\begin{aligned} Info\ Gain_a &= H(c) - H\left(\frac{c}{a}\right) \\ H(c) &= -\sum_{c \in C} p(c) * \log_2 p(c) \\ H\left(\frac{c}{a}\right) &= -\sum_{a \in A} p(a) * \sum_{c \in C} p\left(\frac{c}{a}\right) * \log_2 p\left(\frac{c}{a}\right) \end{aligned} \quad (1)$$

Where H (X) is entropy X, and p (X) is the probability of X

2.3 Building Neural Network Scheme

In this study a machine learning model in the form of artificial neural network backpropagation was formed with architecture as presented in Table 3.

Table.3 Neural Network Scheme

No	Net Layer	Neuron Numbers	Activation Function
1	Input	(according to selected feature)	
2	Hidden	2n+1 (“n” according to selected input feature)	Logsig
3	Output	2	Purelin

The use of a hidden layer in neural network architecture is based on the reason that a hidden layer is sufficient to solve the classification problem [15], and the number of hidden layer neurons is 2n where n is the number of input layer neurons [16]. The function used to train neural networks is determined using the quasi newton method (in Matlab trainlm) which is able to provide divergence speed compared to the scaled conjugate or resilient propagation method [17].

3. Results and Discussion

The experiments in this study were carried out with Matlab 2015B software running on a Windows 10 64bit operating system platform. The results of the feature selection stages of the UNSW-NB15 dataset produce a sequence of features as presented in Table 4.

Table.4 Feature Selection Result

No	Category	Order of Feature	Remarks
1	All Feature	2, 6, 9, 10, 12, 1, 3, 7, 5, 11, 15, 4, 14, 8, 13	15 Feature (all feaatures, no selection)
2	Selected 5	2, 6, 9, 10, 12	5 Feature Selected
3	Selected 7	2, 6, 9, 10, 12, 1, 3	7 Feature Selected
4	Selected 9	2, 6, 9, 10, 12, 1, 3, 7, 5	9 Feature Selected

In this study, four feature schemes are used as input from artificial neural network classifiers to determine the effectiveness of training and classification accuracy resulting from the feature selection process. Based on the input schemes from the feature selection, four different neural network architecture schemes were formed, referring to Table 3. Four neural network architectural schemes related to input features are presented in Table 5.

Table.5 Neural Network Scheme Regarding Input Feature Selection

Scheme Number	Input Layer Neuron Numbers	Hidden Layer Neuron Numbers	Output Neuron Numbers
1	15	30	2 (Normal & DDoS)
2	5	10	2 (Normal & DDoS)
3	7	14	2 (Normal & DDoS)
4	9	18	2 (Normal & DDoS)

The four neural network schemes were trained with the same parameters namely epoch = 20,000; momentum = 0.95; learning rate = 0.1; goal = 0.01; performance evaluation = mean-squared error; gradient = 0.01e-10; mu = 1.00e + 10. The amount of data in the dataset with a total of 1200 lines is randomly divided into three blocks, namely training, testing, and validation. The distribution of dataset blocks is done randomly using the default Matlab dividerand function which produces 70% of training data, 15% of testing data, and 15% of validation data. The results of the training of four neural network schemes related to feature input are presented sequentially in Fig 1 to Fig 4. A summary of the performance of the training results and accuracy of the four neural network scheme models is presented in Table 6.

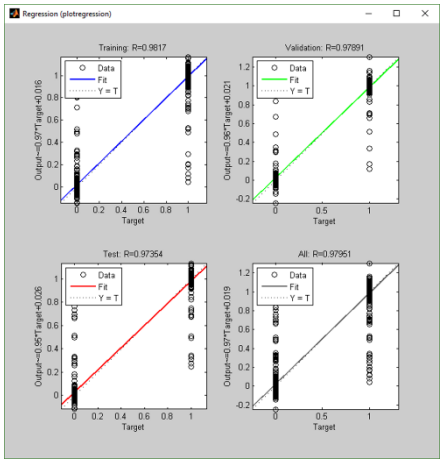

Fig. 1. Results of the 1st scheme of neural network training

Neural network training with architectural schemes 15-30-2 produces an overall regression value of 0.979510 as presented in Fig 1.

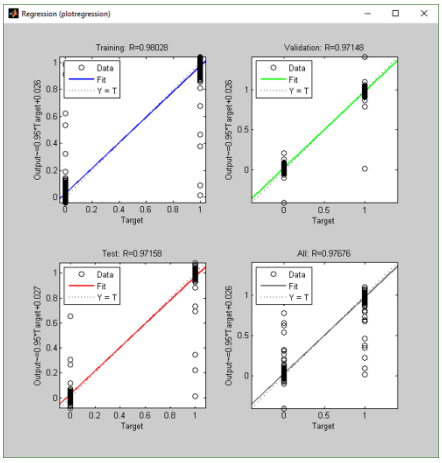

Fig. 2. Results of the 2nd scheme of neural network training

Neural network training with architecture schemes 5-10-2 (5 input feature selection) produces an overall regression value of 0.976760 as presented in Fig 2.

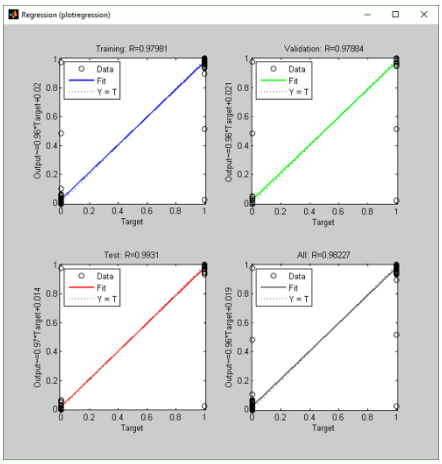

Fig. 3. Results of the 3rd scheme of neural network training

Whereas neural network training with architecture schemes 7-14-2 (7 input feature selection) produces an overall regression value of 0.982270 as presented in Fig 3.

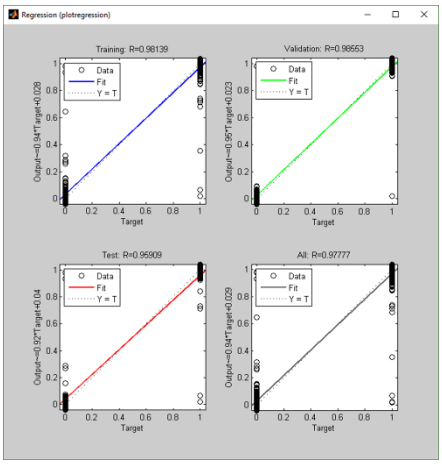

Fig. 4. Results of the 4th scheme neural network training

While neural network training with 9-18-2 architecture schemes (9 input feature selection) produces an overall regression value of 0.977770 as presented in Fig 4.

Table.6 Performance and Accuracy From Four Neural Network Scheme Regarding Input Feature Numbers

Net Scheme Number	Feature Input	Architecture	Regression Result	Epoch Result	Mean Squared Error Result	Accuracy Result
1	15 (all features use as input)	(15-30-2)	0.979510	1166	0.015443	96.12%
2	5 selected	(5-10-2)	0.976760	953	0.012822	95.85%
3	7 selected	(7-14-2)	0.982270	429	0.009011	97.76%
4	9 selected	(9-18-2)	0.977770	752	0.011841	95.33%

In summary, the results of the accuracy of each neural network scheme related to the number of feature selection inputs are presented in Fig. 5.

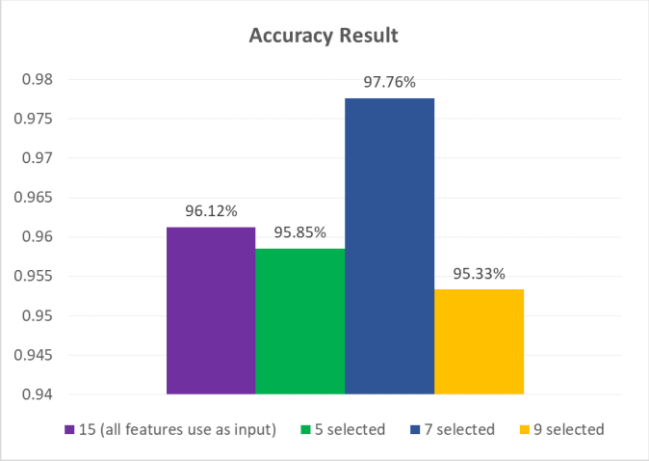

Fig. 5. Accuracy result regarding neural network selected input scheme

4. Conclusion

Based on the results of experiments that have been carried out, it is found that feature selection plays an important role in the accuracy of detection results and the efficiency of machine learning training in classification problems. In this study, the combination of seven main features of the dataset used as an input neural network classifier namely feature number 2, 6, 9, 10, 12, 1, and 3 produces the highest accuracy value of 97.76% compared to the three other feature combination schemes, namely 15 feature input schemes, 5 feature input schemes, and 9 feature input schemes. The seven feature combination scheme also produces a neural network model that has the best training efficiency, which

is characterized by the smallest epoch and mean-squared error among other schemes, namely 429 epochs and 0.009011 mean-squared errors. In contrast, the validation regression value of the neural network model with the input of seven selection features, produces the largest value of 0.982270, which means that the neural network model provides a high match between input and training targets. In the end it can be concluded that to cover the ordinary IDS deficiency in solving DDoS attack detection problems, based on the UNSW-NB15 dataset and neural network backpropagation classifier, seven selected features are needed from the fifteen available features. The seven features are able to produce an accuracy of 97.76% and training classifier efficiency of 429 epochs.

Acknowledgment

Thanks to Dr Nour Mustafa from the University of New South Wales who was pleased to share the UNSW-NB15 network traffic dataset as the main basis of this research.

References

[1] S. Meysam, T. Nezhad, M. Nazari, and E. A. Gharavol, "A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks," *EEE Commun. Lett.*, vol. 20, no. 4, pp. 700–703, 2016.

[2] M. Indra, W. Pramana, Y. Purwanto, and F. Y. Suratman, "DDoS Detection Using Modified K-Means Clustering with Chain Initialization Over Landmark Window," in *IEEE 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2015, pp. 7–11.

[3] W. Fuertes, A. Tunala, R. Moncayo, F. Meneses, and T. Toulkeridis, "Software-based Platform for Education and Training of DDoS Attacks using Virtual Networks," *2017 Int. Conf. Softw. Secur. Assur.*, pp. 94–99, 2017, doi: 10.1109/ICSSA.2017.19.

[4] F. Z. Chowdhury, "Economic Denial of Sustainability (EDoS) Mitigation Approaches in Cloud : Analysis and Open Challenges," in *IEEE International Conference on Electrical Engineering and Computer Science (ICECOS) 2017 Economic*, 2017, pp. 206–211.

[5] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, 2016, doi: 10.1109/TC.2016.2519914.

[6] J. Zhang, P. Liu, J. He, and Y. Zhang, "A Hadoop based analysis and detection model for IP Spoofing typed DDoS attack," in *2016 IEEE TrustCom-BigDataSE-ISPA*, 2016, pp. 1978–1985, doi: 10.1109/TrustCom.2016.300.

[7] G. Ramadhan, Y. Kurniawan, C. Kim, A. T. C. P. Syn, and F. Ddos, "Design of TCP SYN Flood DDoS Attack Detection Using Artificial Immune Systems," in *IEEE 6th International Conference on System Engineering and Technology*, 2016, pp. 72–76.

[8] P. Machaka and A. Bagula, "Using Exponentially Weighted Moving Average Algorithm to Defend Against DDoS Attacks," in *IEEE 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics*, 2016.

[9] S. Hajar *et al.*, "A Neural Network Model for Detecting DDoS Attacks Using Darknet Traffic Features," in *IEEE 2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, no. November 2014, pp. 2979–2985.

[10] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[11] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.*, vol. 25, no. 1–3, pp. 18–31, 2016, doi: 10.1080/19393555.2015.1125974.

[12] N. M. A. Moustafa, "Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic," no. November, 2017.

[13] K. K. Vasan and B. Surendiran, "Feature subset selection for intrusion detection using various rank-based algorithms," *Int. J. Comput. Appl. Technol.*, vol. 55, no. 4, p. 298, 2017, doi: 10.1504/ijcat.2017.086017.

[14] S. Khan, A. Gani, A. W. A. Wahab, and P. K. Singh, "Feature Selection of Denial-of-Service Attacks Using Entropy and Granular Computing," *Arab. J. Sci. Eng.*, vol. 43, no. 2, pp. 499–508, 2018, doi: 10.1007/s13369-017-2634-8.

[15] M. Anthony *et al.*, *Neural Network Learning: Theoretical Foundations*. Edinburgh, Scotland: Cambridge University Press, 2009.

[16] I. Riadi, A. Wirawan, and S. -, "Network Packet Classification using Neural Network based on Training Function and Hidden Layer Neuron Number Variation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 248–252, 2017, doi: 10.14569/ijacsa.2017.080631.

[17] I. Riadi, A. W. Muhammad, and Sunardi, "Neural network-based ddos detection regarding hidden layer variation," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 15, pp. 3684–3691, 2017.