Network Packet Classification Using Neural Network Based on Training Function and Hidden Layer Neuron Number Variation

By Imam Riadi

WORD COUNT	3022	TIME SUBMITTED	16-NOV-2018 08:06AM
		PAPER ID	42008868

Network Packet Classification using Neural Network based on Training Function and Hidden Layer Neuron Number Variation

Imam Riadi
Department of Information System
Ahmad Dahlan University
Yogyakarta,
Indonesia

Arif Wirawan Muhammad
Department of Information Technology
Ahmad Dahlan University
Yogyakarta, Indonesia

Sunardi
Department of Electrical Engineering
Ahmad Dahlan University
Yogyakarta,
Indonesia

***Abstract*—Distributed denial of service (DDoS) is a structured network attack coming from various sources and fused to form a large packet stream. DDoS packet stream pattern behaves as normal packet stream pattern and very difficult to distinguish between DDoS and normal packet stream. Network packet classification is one of the network defense system in order to avoid DDoS attacks. Artificial Neural Network (ANN) can be used as an effective tool for network packet classification with the appropriate combination of numbers hidden layer neuron and training functions. This study found the best classification accuracy, 99.6% was given by ANN with hidden layer neuron numbers stated by half of input neuron numbers and twice of input neuron numbers but the number of hidden layers neuron by twice of input neuron numbers gives stable accuracy on all training function. ANN with Quasi-Newton training function doesn't much affected by variation on hidden layer neuron numbers otherwise ANN with Scaled-Conjugate and Resilient-Propagation training function.**

***Keywords*—Classification; DDoS; neural; network; training; function; hidden; layer**

I. Introduction

Distributed denial of service (DDoS) is a structured network attack coming from various sources and fused to form a large packet stream. DDoS attacks, generally utilizing resources from the slave computer coordinated by the attacker to decrease the target network resources causing legitimate client cannot access these resources. DDoS packet stream pattern behaves as normal packet stream pattern and it is very difficult to distinguish between DDoS and normal packet stream [1].

DDoS packet stream with a large volume causes the target system cannot handle and end up with a loss of resources such as system shutdown, loss of data, moreover, the system loses the overall of owned services [2], [3]. Network packet classification is one of network defense system in order to avoid DDoS attacks [4]. Network packet classification can be carried out by utilizing Artificial Neural Network (ANN) method.

Network packet classification for DDoS attacks detection in TOR network using ANN carried on research [5] by utilizing optimization of a sinusoidal function as a feature extractor of the network packet. ANN used in [6] with Resilient-Backpropagation function combined with the ensemble of classifier outputs method and Neyman-Pearson cost minimization strategy for detection of DDoS attack based on DARPA and KDDCUP datasets. Research [7] adopted the ANN method to detect DDoS attacks based on darknet traffic. TCP/80 and UDP/53 packets used as input and optimized by Locally Sensitive Hashing methods. ANN used in [8] to recognize illegal packets in the network, by taking advantage of the Backpropagation functions. TCP, ICMP, and UDP packet used as inputs in the [8]. Research [9] proved that the ANN method can be used to detect a new type of DDoS attack, in Hadoop and HBase environment.

Based on earlier research regarding packet classification with ANN, this study focuses on the ANN training function to find out the best training function layer for packet classification. DDoS dataset published by the Center for Applied Internet Data Analysis (CAIDA) and network normal dataset published by Ahmad Dahlan University Networks Laboratory are used in this study.

II. Packet Classification Approach

The study of packet classification with artificial neural network applying variation of training function and hidden layer neuron number, involves steps as seen on Fig. 1.

1) Get network DDoS dataset and Normal dataset from CAIDA and Ahmad Dahlan University Networks Laboratory in .pcap format.

2) Extract network packet using statistical method to get network features, that are average packet size, number of packets, time interval variance, packet size variance, packet rate, and number of bytes.

3) Train ANN with three training function (Quasi-Newton, Resilient-Propagation, Scaled-Conjugate).

4) Classification result comparison using accuracy, mean-squared error (mse), and iteration parameters.

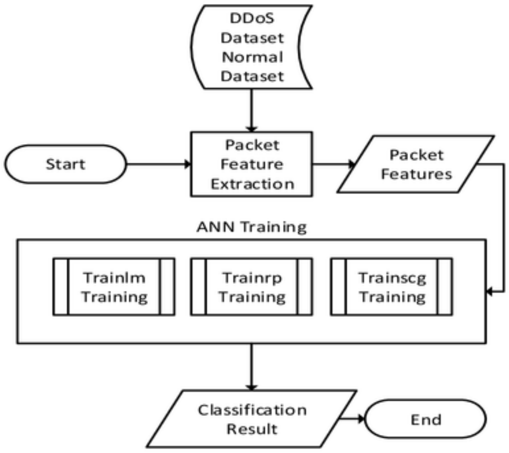

Fig. 1. Steps of network packet clasification.

Accuracy is the ratio between the addition of normal and DDoS packet that is recognized by the system and compared to the overall packet data. Mean square error (mse) is the most ANN important parameter for performance evaluation of training functions parameters [10]. Mean square error reflects an absolute error of ANN training output pattern with desired output pattern. The iterations reflect the time taken by ANN to reach convergence also a tradeoff indicator between training time and convergence.

III. ANN Components Framework

A. Network Packet Features

To classify the network packet, the first step is extracted from the network feature of the dataset. The aim of feature extraction is to measure certain attributes in original data that distinguish one input pattern from another pattern. In this study, network packet stream extracted to six features based on statistical method. Those six features are:

1) *Average packet size*: The longer DDoS attack occurs, then it is always followed by a rise in the value of average packet size [11].

2) *Number of packets*: DDoS attacks overwhelm a target computer network by sending many packets at a certain time lag. DDoS always result in high number of the packet [11].

3) *Time interval variance*: DDoS attack delivers packages in large numbers occurred in a certain time span, the value of time interval variance will be smaller and nearly zero. Time interval variance stated as (1) [12].

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

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

4) *Packet size variance*: The normal traffic resulting high packet size variance values within DDoS attacks resulting close to zero packet size variance value, due to the monotony packet size that sent to target. Packet size variance stated as (2) [12].

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

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

5) *Packet rate*: Packet rate reflects the number of packets sent by the source address to a destination address within a specific time frame as stated on (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.

6) *Number of bytes*: DDoS attack always increases the number of bytes in constant [12].

B. Training Function

There are numbers of batch training algorithms which can be used to train an Artificial Neural Network [13]. The most used training algorithms are:

1) Newtonian training function is fast to reach convergence than conjugate gradient methods, but Newton's method is complex and time-consuming to compute the Hessian matrix for feed forward neural networks [14], [15]. Based on Newton's method there a new class of method is called a Quasi-Newton method (Matlab trainlm) which doesn't require calculation of second derivatives. The Quasi-Newton method updates a Hessian matrix in each iteration of the algorithm [16], [17].

2) Resilllent-Propagation training function (Matlab trainrp) refers to the gradient-descent algorithm that removes the effect of partial derivative magnitude from the activation function. In this case a partial derivative of the activation function is used to determine the direction of the neural network weights, whereas the magnitude of the partial derivatives has no effect on the weight changes. So that the weight changes of the neural network can become more stable in achieving the minimum gradient [15].

3) Scaled-Conjugate training function (Matlab trainscg) refers to the conjugate-gradient algorithm that exploits the gradient's negative direction to match the weight changes of the neural network layer so that it affects the number of iterations the neural network takes to achieve convergence [15].

C. ANN Layer Scheme

There is no certainty that the best number of neurons and hidden layers are used to resolve a problem with an ANN [18]. Based on that reason, this study does some variation on hidden layer neuron numbers as seen on Table 1.

TABLE. I. Hidden Layers Variation

Type	Input Neurons	Hidden Layer Neuron Variations	Output Neurons
1.	6	3	2
2.	6	6	2
3.	6	12	2
4.	6	13	2

D. Comparison Parameters

Accuracy, mean-squared error, and iteration parameters was used in this research for classification performance analysis.

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

2) Mean-squared error (mse), reflect an absolute error of ANN training actual output pattern with desired output pattern.

3) Iteration reflect the time that takes by ANN to reach convergence [16].

IV. Result

Experiments were carried out on Matlab 2010R environment running on Windows 7 64-bit. Experimental dataset consists of 500 DDoS traffic data and 500 Normal traffic data by six features. In purpose of ANN training, dataset was divided by default on Matlab 2010R into 70% sets for training, 15% sets for validation, and 15% sets for testing. Distribution of dataset for training, validation, and testing was created by random function (Matlab dividerand) to avoid the bias tendency in the sample pattern.

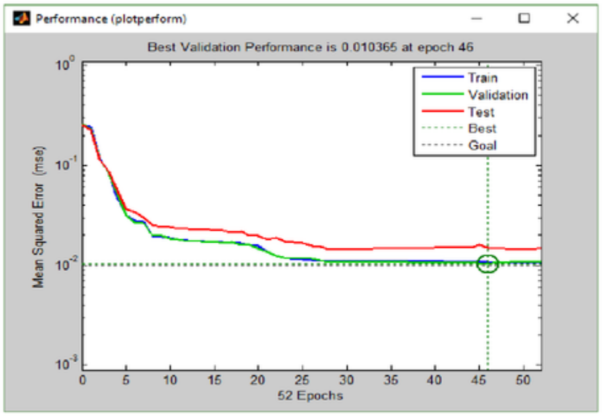

Fig. 2. Quasi-Newton Training Result on Layer 6-(3)-2.

Sigmoid transfer function used in the hidden and output layer. The basic parameters epoch = 20000, performance function = mse, goal = 0.01, maximum fail = 6, minimum gradient = 1.00e-10, mu = 1.00e+10 were used in the training process. For simplification purpose, the result for each training function displayed only for ANN layer 6-(3)-2. Quasi-Newton method (Matlab trainlm) training result for ANN layer 6-(3)-2 presented on Fig. 2. Scaled-Conjugate method (Matlab trainscg) training result for ANN layer 6-(3)-2 presented on Fig. 3.

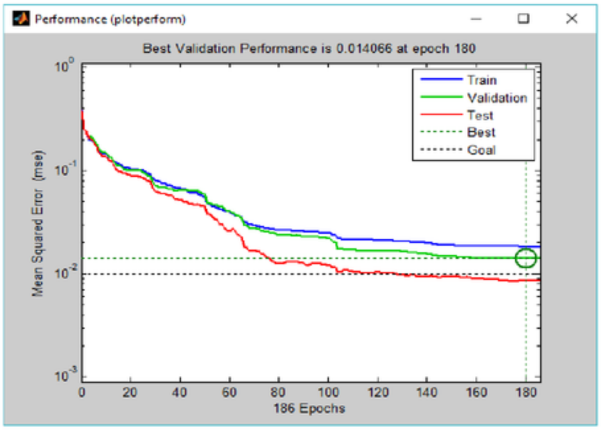

Fig. 3. Scaled-Conjugate Training Result on Layer 6-(3)-2.

Resilient-Propagation method (Matlab trainrp) training result for ANN layer 6-(3)-2 presented on Fig. 4.

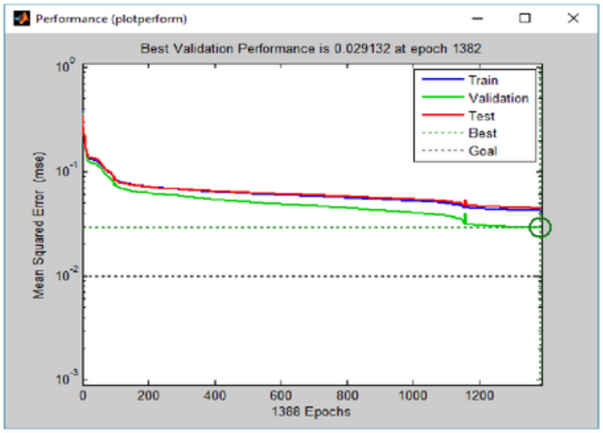

Fig. 4. Resilient-Propagation Training Result on Layer 6-(3)-2.

All training result stated that there was no overtraining faced on ANN scheme.

E. Accuracy

Quasi-Newton training function (Matlab trainlm) resulted stable accuracy value against all ANN layer schemes as stated on Fig. 5. The highest accuracy value 0.996 (99.6%) was achieved by ANN with Scaled-Conjugate training function (Matlab trainscg) under 6-(3)-2 layer scheme and also ANN with Quasi-Newton training function (Matlab trainlm) under 6-(12)-2 layer scheme. However, the Scaled-Conjugate training function (Matlab trainscg) resulted less consistent value on other ANN layer schemes. Based on Fig. 5 best classification accuracy was given by ANN with the number of hidden layer neurons by 1/2n and 2n. Where, n is the number of input nurons.

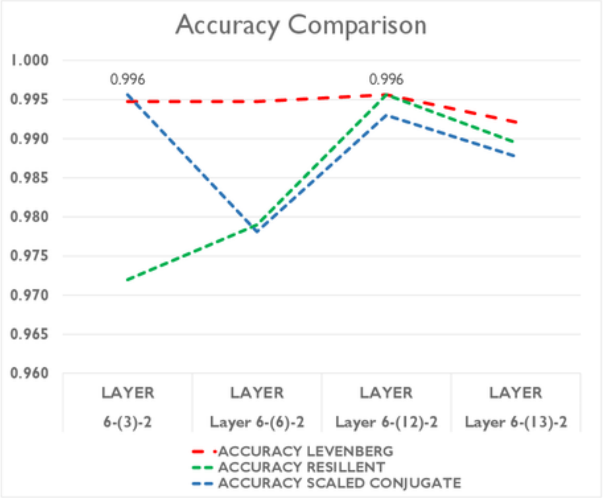

Fig. 5. Accuracy comparison.

The number of hidden layers neuron by 2n gives stable accuracy on all training function, as compared to Kolmogorov's theory that stated the best number of hidden layer neurons to solve ANN problem is 2n + 1 which produce accuracy value that tends to be low on this experiments.

F. Mean-Squared Error

As stated From Fig. 6, the conclusion that can be drawn is as follows:

1) Quasi-Newton (Matlab trainlm) training function resulted small average mse value on all ANN layer schemes compared to the Scaled-Conjugate (Matlab trainscg) and Resilient-Propagation (Matlab trainrp) training functions.

2) The number of neurons in the hidden layer don't have a significant effect on MSE value for Quasi-Newton (Matlab trainlm) training functions.

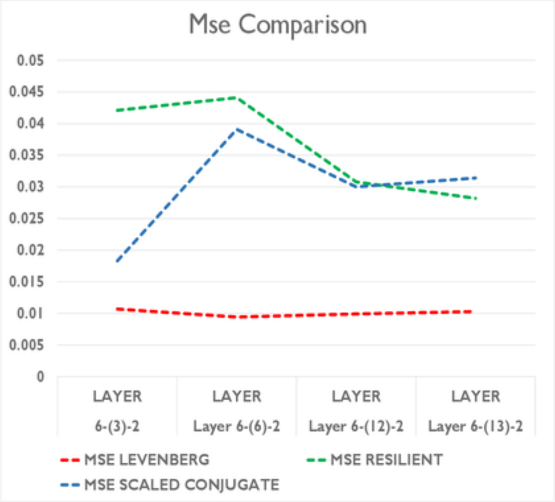

Fig. 6. MSE comparison.

3) The number of neurons in the hidden layer have a significant effect on MSE value for Scaled-Conjugate (Matlab trainscg) and Resilient-Propagation (Matlab trainrp) training functions.

4) More number of neurons in hidden layer can reduce MSE value for Resilient-Propagation (Matlab trainrp) training functions.

5) More number of neurons in hidden layer otherwise increases MSE value on Scaled-Conjugate (Matlab trainscg) training functions.

G. Iteration

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. ANN with Quasi-Newton (Matlab trainlm) training function is fast to reach convergence than conjugate gradient methods and efficient in time. Fig. 7 stated that the number of neurons in the hidden layer don't have a significant effect on ANN convergence speed for Quasi-Newton (Matlab trainlm) and Scaled-Conjugate (Matlab trainscg) training functions. However, the Resilient-Propagation (Matlab trainrp) training function affected with the number of neurons in the hidden layer, more number of neurons in hidden layer can increase convergence speed.

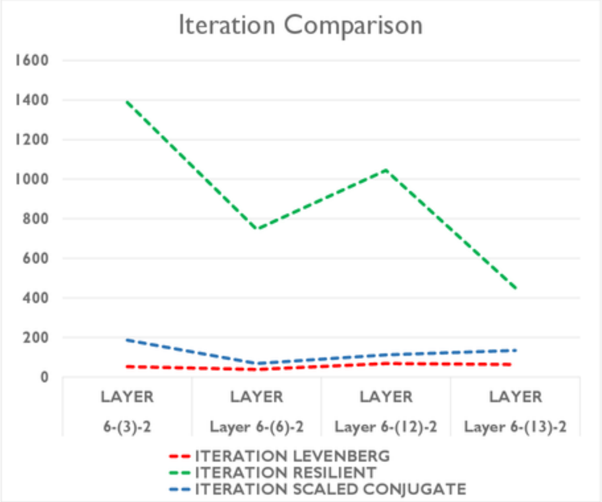

Fig. 7. Iteration comparison.

V. Future Work and Conclusion

Artificial neural network can be used as an effective tool for network packet classification with the appropriate combination hidden layer and training functions. This study found best classification accuracy (99.6%) was given by ANN with the number of hidden layer neurons by 1/2n for Matlab trainscg training function and 2n for Quasi-Newton (Matlab trainlm) training function. Where, n is the number of input neurons. The number of hidden layers neuron by 2n gives stable accuracy on all training function. Quasi-Newton (Matlab trainlm) training function is fast to reach convergence and doesn't much affected by number of hidden layer neurons variation. The significant differences on MSE value is found by applying variation of hidden layer neurons numbers in the neural network trained by Scaled-Conjugate and Resilient-Propagation training function. More number of neurons in hidden layer can reduce MSE value for Resilient-Propagation (Matlab trainrp) training functions and more number of neurons in hidden layer otherwise increases MSE value on Scaled-Conjugate (Matlab trainscg) training functions. In this study, the best suitable number of neurons in hidden layer is 2n, because it gives stable accuracy on all training function.

The results obtained from this study can be used as a basic reference to determine the effective number of hidden layers neuron in building a network packet classification system based on artificial neural network. Further, the study will be improved on other parameters like increasing the sample size of input patterns presented to the network, reducing error goal and use more training method.

References

[1] Mahadev, V. Kumar, and K. Kumar, “Classification of DDoS Attack Tools and its Handling Techniques and Strategy at Application Layer,” 2016 2nd Int. Conf. Adv. Comput. Commun. Autom., pp. 1–6, 2016.

[2] S. H. A. Ali, S. Ozawa, T. Ban, J. Nakazato, and J. Shimamura, “A Neural Network Model For Detecting DDoS Attacks Using Darknet Traffic Features,” 2016 Int. Jt. Conf. Neural Networks, no. November 2014, pp. 2979–2985, 2016.

[3] A. Iswardani and I. Riadi, “Denial of Service Log Analysis Using Density K-Means Method,” J. Theor. Appl. Inf. Technol., vol. 83, no. 2, pp. 299–302, 2016.

[4] I. Riadi, A. W. Muhammad, and Sunardi, “Neural Network-Based DDoS Detection Regarding Hidden Layer Variation,” J. Theor. Appl. Inf. Technol., vol. 95, pp. 1–9, 2017.

[5] T. Ishitaki, D. Elmazi, Y. Liu, T. Oda, L. Barolli, and K. Uchida, “Application of Neural Networks for Intrusion Detection in Tor Networks,” Proc. - IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2015, pp. 67–72, 2015.

[6] M. Kale and D. . Choudhari, “DDOS Attack Detection Based on an Ensemble of Neural Classifier,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 14, no. 7, pp. 122–129, 2014.

[7] S. H. A. Ali, S. Ozawa, T. Ban, J. Nakazato, and J. Shimamura, “A Neural Network Model for Detecting DDoS Attacks using Darknet Traffic Features,” 2016 Int. Jt. Conf. Neural Networks, no. November 2014, pp. 2979–2985, 2016.

[8] 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.

[9] T. Zhao, D. C. T. Lo, and K. Qian, “A Neural Network Based DDoS Detection System Using Hadoop and HBase,” Proc. - 2015 IEEE 17th Int. Conf. High Perform. Comput. Commun. pp. 1326–1331, 2015.

[10] H. Demuth, Neural Network Toolbox Users Guide, Sixth Ed., vol. 24, no. 1. Natick, Massachuset: The MathWorks, Inc, 2002.

[11] C. J. Hsieh and T. Y. Chan, “Detection DDoS Attacks Based On Neural-Network Using Apache Spark,” 2016 Int. Conf. Appl. Syst. Innov. 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,” Int. J. Adv. Res. Comput. Eng. Technol., 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,” Natl. Chin-Yi Univ. Technol. Taichung, Taiwan, pp. 1–4, 2015.

Network Packet Classification Using Neural Network Based on Training Function and Hidden Layer Neuron Number Variation

ORIGINALITY REPORT

9%

SIMILARITY INDEX

PRIMARY SOURCES

1 Teng Zhao, Ming Li, Gabriel Chen, Ying Wang. "Autonomous Vehicle Tracking Control Using Deep Learning and Stereo Vision", 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2018
Crossref
21 words — 1%

2 Haughey, Hamish, Gregory Epiphaniou, and Haider M Al-Khateeb. "Anonymity networks and the fragile cyber ecosystem", Network Security, 2016.
Crossref
20 words — 1%

3 link.library.deakin.edu.au
Internet
19 words — 1%

4 rd.springer.com
Internet
17 words — 1%

5 Chauhan, S., and R. K. Shrivastava. "ESTIMATING REFERENCE EVAPOTRANSPIRATION FROM LIMITED CLIMATIC DATA USING ARTIFICIAL NEURAL NETWORKS", ISH Journal of Hydraulic Engineering, 2009.
Crossref
17 words — 1%

6 documents.mx
Internet
17 words — 1%

7 zdoc.site
Internet
16 words — 1%

8 Sayidul Morsalin, Khizir Mahmud, Humaira Mohiuddin, Md.

Reduanul Halim, Prattay Saha. "Induction motor inter-turn fault detection using heuristic noninvasive approach by artificial neural network with Levenberg Marquardt algorithm", 2014 International Conference on Informatics, Electronics & Vision (ICIEV), 2014
Crossref
16 words — 1%

9 Robinson, R R Rejimol, and Ciza Thomas. "Ranking of machine learning algorithms based on the performance in classifying DDoS attacks", 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2015.
Crossref
16 words — 1%

10 Otto Schwelb. "Series-coupled microring resonator filters with embedded semiconductor optical amplifiers", Microwave and Optical Technology Letters, 09/05/2004
Crossref
15 words — < 1%

11 www.ijccer.org
Internet
15 words — < 1%

12 Jakob Borrits Sabra, Hans Jorgen Andersen, Kasper Rodil. "Hybrid Cemetery Culture: Making Death Matter in Cultural Heritage Using Smart Mobile Technologies", 2015 International Conference on Culture and Computing (Culture Computing), 2015
Crossref
14 words — < 1%

13 Xiaoyan Zhu, Xiaomei Yang, Chenzhen Ying, Guangtao Wang. "A new classification algorithm recommendation method based on link prediction", Knowledge-Based Systems, 2018
Crossref
13 words — < 1%

14 Silvia Bravo, David Mauricio. "DDoS attack detection mechanism in the application layer using user features", 2018 International Conference on Information and Computer Technologies (ICICT), 2018
Crossref
13 words — < 1%

15 ijarcsse.com
Internet
10 words — < 1%

16	Bingcheng Zhu, Julian Cheng, Yongjin Wang, Jun Yan, Jinyuan Wang. "Three-Dimensional VLC Positioning Based on Angle Difference of Arrival With Arbitrary Tilting Angle of Receiver", IEEE Journal on Selected Areas in Communications, 2018 Crossref	9 words — < 1%
17	www.mtome.com Internet	9 words — < 1%
18	shura.shu.ac.uk Internet	8 words — < 1%
19	M. Klima, P. Pata, K. Fliegel, P. Hanzlik. "Subjective image quality evaluation in security imaging systems", Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology, 2005 Crossref	8 words — < 1%
20	Kybernetes, Volume 42, Issue 7 (2013-11-30) Publications	7 words — < 1%
21	Lefik, M.. "Artificial Neural Networks in numerical modelling of composites", Computer Methods in Applied Mechanics and Engineering, 20090501 Crossref	7 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		