Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 3, March 2019, pp. 919~926
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i3.pp919-926

Anomaly-based intrusion detector system using restricted growing self organizing map

Tomi Yahya Christyawan, Ahmad Afif Supianto, Wayan Firdaus Mahmudy
Faculty of Computer Science, Brawijaya University, Malang, Indonesia

Article Info

Article history:

Received Jul 06, 2018
Revised Nov 11, 2018
Accepted Dec 8, 2018

Keywords:

Big data
Clustering reference vector
Growing som
Intrusion Detector System
Self-organizing map

ABSTRACT

The rapid development of internet and network technology followed by malicious threats and attacks on networks and computers. Intrusion detection system (IDS) was developed to solve that problems. The development of IDS using machine learning is needed for classifying the attacks. One method of the classification is Self-Organizing Map (SOM). SOM able to perform classification and visualization in learning process to gain new knowledge. However, the SOM has less efficient in learning process when applied in Big Data. This study proposes Restricted Growing SOM method with clustering reference vector (RGSOM-CRV) and Parallel RGSOM-CRV to improve SOM efficiency in classification with accuracy consideration to solve Big Data problem. Growing process in RGSOM is restricted by maximum nodes and growing threshold, the reupdate weight process will update unused reference vector when map size already maximum, these two processes solve the consuming time of regular GSOM. From the results of this research against KDD Cup 1999 dataset, proposed method Parallel RGSOM-CRV able to give 91.86% accuracy, 20.58% false alarm rate, 95.32% recall or detection rate, and precision is 94.35% and time consuming is outperform than regular Growing SOM. This proposed method is very promising to handle big data problems compared with other methods.

Corresponding Author:

Tomi Yahya Christyawan,
Faculty of Computer Science,
Brawijaya University,
Veteran street, Malang, Indonesia.
Email: tomi@rekavisitama.net

1. INTRODUCTION

The security threat to internet usage and computer networks is increasing. Several types of new attacks on the network appear periodically, this make a chalange to develop a flexible and adaptive network security. Developing techniques to detect anomaly-based network intrusion to protect a computer system and network from malicious activity attacks call Intrusion Detection System (IDS), as the detection of suspicious network traffic and computer usage can not be done by conventional firewalls.

Some development of IDS based on machine learning technique. The methods used for anomaly-based intrusion detection are commonly differentiated into classification and clustering. However, there is also a hybridization between clustering and classification for intrusion detection system. In the classification method, some studies use single classifier such as KNN [1], Support Vector Machine (SVM) [2], artificial neural network [3-6] to solve IDS problem. Other researchers use hybrid methods of heuristic algorithm with classifier method [7-9], Multi-level SVM and Extreme Learning Machine with K-Mean [10], Decision Tree and SVM [11], Tree Augmented Naïve Bayes (TAN) and Reduced Error Pruning (REP) [12]. IDS-related studies using clustering include K-Mean, K-Medoids, A-SPOT [13], CANN [14].

One method of classification and reduction data that can visualize the learning process is SOM. In some studies SOM able to solve the problem of classification with better result [15]. However, SOM

experienced constraints in efficiency during the process of learning with large data (Big Data), this have high time consuming. The problem is due to the characteristics of SOM that calculate the distance between input vector and reference vector to decide the winning neuron on the hidden layer for each epoch. Big data problems seem to be faced by Li at al. [1], they just select 5,552 instances from KDD Cup 99 sample data as the training data, and 5,552 instances as testing data for their experiment, not from the entire KDD Cup 99 data.

Based on research conducted by Alahakoon [16], Growing Self Organizing Map (GSOM) can be used to build reference vector gradually based on training data. However, this able to solve the problem in the first epoch only, in next epoch, topological map will be larger, and the problem of time consume will reappear on next epoch. This study proposes Restricted Growing Self Organizing Map with clustering reference vector (RGSOM-CRV) and Parallel RGSOM-CRV to solve the issue on regular GSOM to handle big data problems. This research will measure RGSOM-CRV and Parallel RGSOM-CRV efficiency of time consuming and accuracy, false alarm rate, precision, and recall.

2. RESEARCH METHOD

This research procedure shown in Figure 1. Features from dataset will be selected according to selected features, and each feature value will be normalizing before process at RGSOM-CRV. Data train and data testing will be treat same way before process in RGSOM-CRV.

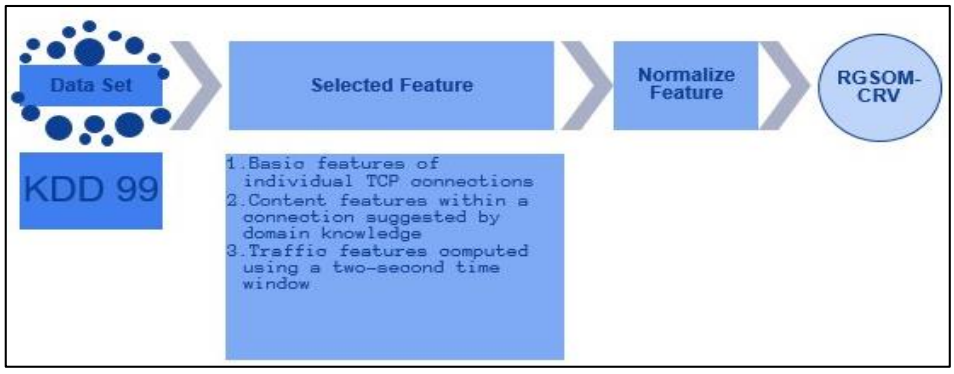

Figure 1. Research procedure

2.1. KDD Cup 99 Dataset

Dataset KDD Cup 99 from UCI separate with 10% data train (494,021 instances) and data test (4,898,431 instances), it's have 41 features. This dataset category in 4 attacks class dos, probe, r2l, u2r and normal category. This research will use all data provided by this dataset to measure the efficient and effective of proposed method to handle big data problems.

2.2. Selected Feature

Data train will be processes with selected features. According to KDD Cup 99 task (http://kdd.ics.uci.edu/databases/kddcup99/task.html) which adapted from Stolfo et al paper [17], there are three types feature selection, basic features of individual TCP connection, content features within a connection suggested by domain knowledge, and traffic feature computed using a two-second time window. In this research basic features of individual TCP connection category type be chosen. Features used in this type are duration, protocol type, service, flag, src bytes, dst bytes, land, wrong fragment, urgent.

2.3. Normalize Feature

After dataset selected by basic feature type, and then this feature will be normalizing with (1) before processed with RGSOM. Where n is normalized value, *x* is real value, argmin (Fi) is smallest value from *i-th* features, argmax (Fi) is largest value from *i-th* features. After normalize data will process in RGSOM-CRV to be train and then test with data test.

$$n = \frac{x - \text{argmin(Fi)}}{\text{argmax (Fi)} - \text{argmin(Fi)}} \quad (1)$$

2.4. GSOM

Growing SOM is modified regular SOM which proposed by Alahakoon [16]. GSOM procedure according to Alahakoon are Initialization phase, Growing phase, and Smoothing phase. Figure 2 show new node generation from the boundary of the network. Figure 2a is initial node using 4 nodes as reference vector. Figure 2b show that high error occurs when winner node distance with reference vector is more than growing threshold. Figure 2c show the growing shcema of GSOM. Growing phase will occur if growing threshold smaller than distance of winner node.

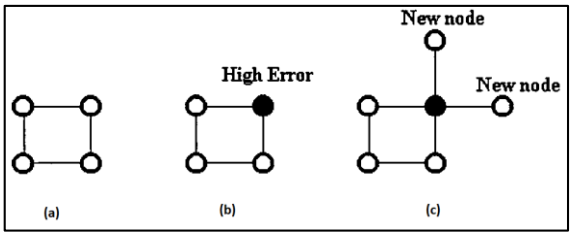

Figure 2. New node generation from the boundary of the network

2.5. RGSOM-CRV

This research propose RGSOM-CRV, this method is extend from regular SOM [15] and regular GSOM [16]. The different with regular GSOM is the growing of reference vector is restricted by maximum size of nodes (MN), and two grid map dimensions. Map will be growing if distance from winning node with input node is more that growing threshold (GT). If the length of nodes in map more that MN, map will stop growing and reupdate by selected randomly one reference vector which never hit (not be winner yet) by input. After select winner node, same with regular SOM, weight of neighborhood will be update. One reference vector could be more than one time selected to be winner node called clustering reference vector (CRV). Clustering reference vector will be a group of input with similarity weight according to minimum GT. CRV method will reduce the size of topographical map, this method can decrease time consuming when select winner node process.

Figure 3 is the flowchart of RGSOM procedure, and Figure 3a is the whole process of RGSOM procedure. In initialization process user need setting the maximum map size and maximum node for the map this propose for restricted the size of the growing node in the map. User also setting the value of start learning rate and stop learning rate, start growing threshold, stop growing threshold, and maxEpoch. Initialization process also generate initial nine reference vector nodes for the map, as shown on Figure 4a.

$$S(t) = S_{start}\left(\frac{S_{end}}{S_{start}}\right)^{\frac{t}{t_{end}}} \quad (2)$$

$$\boldsymbol{win} = argmin\{\|\boldsymbol{x} - \boldsymbol{w_k}\|\} \quad (3)$$

$$\boldsymbol{w_k}(t+1) = \boldsymbol{w_k}(t) + h_{wink}\,(\boldsymbol{x}\,(t) - \boldsymbol{w_k}(t)\,) \quad (4)$$

$$h_{wink} = \alpha(t)\, e^{-\left(\frac{\|win - r_k\|^2}{\sigma^2(t)}\right)} \quad (5)$$

Figure 3b is training process flowchart. Update learning rate and update growing threshold using monotonic decrement function, figure out by (2), where *S(t)* is the updated learning rate or growing threshold, S_{start} is starting value, S_{end} is ending value, *t* is current epoch, t_{end} is max epoch. Learning rate and growing threshold update used to set current learning rate and growing threshold at current epoch. The winner node calculate use regular SOM procedure follow by (3). Where, ***win*** is winner node, ***x*** is the input vector and $\boldsymbol{w_k}$ is *k-th* reference vector. Reference vector are list of nodes generate by first node (nine square node which position is in the center of map Figure 4a) and generate by growing function. If winner distance more than the growing threshold in the current epoch and generated nodes length smaller than maximum node, reference vector will grow with random weight. If generated nodes length more than maximum node, unused reference vector will be reupdated. Unused reference vector are nodes that never hit or not yet selected to be winner. The scheme of the growing of RGSOM shown on Figure 4c. Neighborhood of winning

neuron will be updated use (4), where h_{wink} is gausian neighborhood function decide by (5). Learning rate notation is $\alpha(t)$ and $\sigma(t)$ is neighborhood radius, both are a monotonically decreasing scalar function of t follow the rule of (2), r_k is *k-th* neighborhood of winner node. Selected winner node will be push to the list of used nodes. Figure 3c is testing procedure flowchart. At testing purpose, the winner will be chosen from map generate by training step and used nodes will be selected to be reference vector for testing input. The winner node will be used to calculate True Negative (TN), True Positive (TP), False Negative (FN), False Positive (FP) values. True negative is correct prediction which real category is normal and labeled as normal. True positive is correct prediction which real attack category and labeled as attack. False negative is incorrect prediction which the real category is attack but predicted as normal. False positive is incorrect prediction which the real category is normal but labeled as attack.

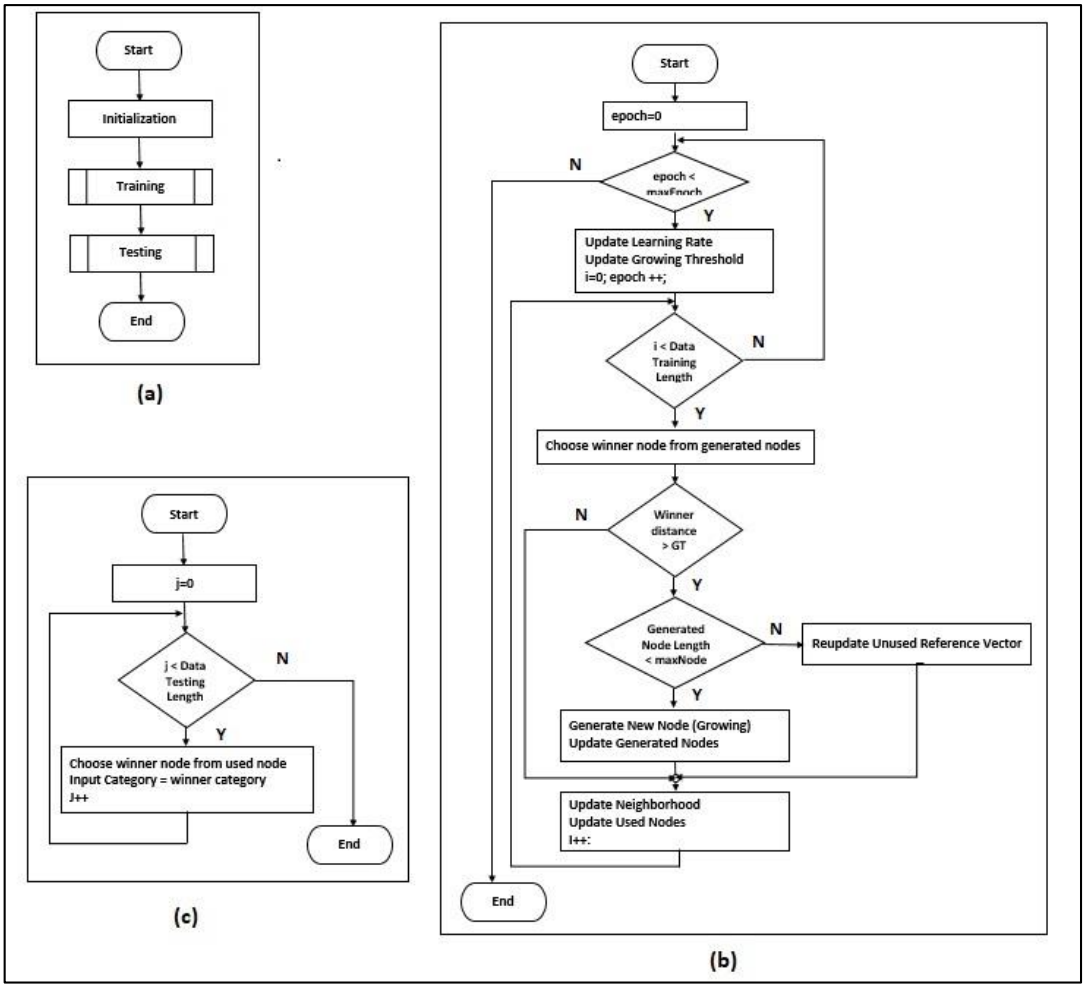

Figure 3. Flowchart RGSOM-CRV Procedure.

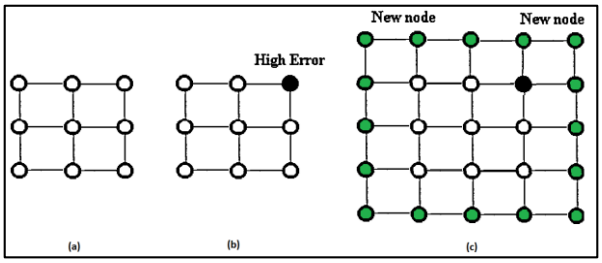

Figure 4. Square growing node schema

2.6. Measurement

This experiment has four measurements to evaluate this method, accuracy (ACC), false alarm rate (FAR), detection rate (DTR) or recall, precision. Accuracy is total correctly classified example to total number of example. Accuracy calculate with (6):

$$\text{ACC} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \text{ x } 100\% \quad (6)$$

False alarm rate is percentage of normal category that have been label as attack to total number of normal examples, and calculate as (7):

$$\text{FAR} = \frac{\text{FP}}{\text{TN+FP}} \text{ x } 100\% \quad (7)$$

Detection rate or recall is standing for correctly label as attacks to total number of attacks, use (8) as formula.

$$\text{DTR or Recall} = \frac{\text{TP}}{\text{TP+FN}} \text{ x } 100\% \quad (8)$$

Precision is being the percentage of correctly label attacks as attacks to total number of instance labeled as attacks, and calculate with (9):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \text{ x } 100\% \quad (9)$$

3. RESULTS AND DISCUSSION

RGSOM-CRV and Parallel RGSOM-CRV in this experiment will initialize with same start learning rate (LRstart), stop learning rate (LRstop), start growing threshold (GTstart), stop growing threshold (GTstop), max epoch, map size. This experiment uses LRstart=0.9, LRstop=0.1, GTstart=0.05, GTstop=0.01, maxEpoch=4, mapSize=100x100. Maximum node for RGSOM-CRV is 5000 and maximum node for Parallel RGSOM-CRV different between protocol, Udp have maximum nodes 3000, and tcp and icmp is 5000, different value of this setting is because udp at training instance much fewer that tcp and icmp as shown at Table 1. This experiment using computer with specification: Processor Intel Core i7-6500U CPU @2.5GHz 2.60 GHz, Memory 8 GB, with system 64-bit Operating System.

Table 2 show the regular GSOM consume more than 2 days and ended in second epoch because have memory limit issue due the map growing bigger. From this experiment, GSOM not capable to handle large data in our experiment due the limitation of memory and time consuming and this is not whorted to be continued.

Table 1. Distribution of Protocol Type

Protocol Type	Count
icmp	283.602
tcp	190.065
udp	20.354

Table 2. GSOM Experiment

	ACC	FAR	DTR/Recall	Precision	Training Time
epoch 1	-	-	-	-	02:32:11
epoch 2	-	-	-	-	More than 2 days and then stopped

The result from five experiments of Parallel RGSOM-CRV show in Table 4. The average Parallel RGSOM-CRV accuracy is 91.86% and false alarm rate is 20.58%, recall or detection rate is 95.32%, and precision is 94.35%. The average time consume while training using Parallel RGSOM-CRV is 6 hours 33 minute and 18 second (four epoch). However, time consuming for testing is 46 minutes and 39 second.

From Table 3 and Table 4 Parallel RGSOM-CRV is outperform than RGSOM-CRV with 91.86% in accuracy, false alarm rate is 20.58%, 95.32% for recall, and 94.35% in precision. However, precision from both experiment have good result. From Table 3 at third experiment, accuracy of RGSOM-CRV have good result than other experiment. This can be happened because RGSOM-CRV is generate randomly and in each experiment, so different result may be obtained. RGSOM-CRV maybe could have different result too for

different GTStart and GTstop setting. The result from the fifth experiment using Parallel RGSOM-CRV for each protocol shown at Table 5. The parallel RGSOM-CRV have total accuracy is 97.27%, false alarm rate is 12.72%, recall or detection rate is 99.79%, and precision is 96.87%. Icmp protocol have largest accuration, false alarm rate, detection rate and largest precission.

Five experiments for both methods will be evaluated. Table 3 show the result of five experiment using RGSOM-CRV. The average RGSOM-CRV accuracy is 51.45% and false alarm rate is 11.80%, recall or detection rate is 41.78%, and precision is 93.09%. The average of time consume training using RGSOM-CRV is 5 hours and 31 minutes and 1 second. Time consume while testing is 1 hour and 44 minutes and 59 second.

Table 3. RGSOM-CRV Experiments Result

	ACC	FAR	DTR/Recall	Precision	Training Time	Testing Time
Experiment 1	38.22%	18.08%	24.52%	81.21%	05:17:24	01:39:05
Experiment 2	38.62%	18.69%	27.94%	85.66%	05:25:42	01:32:56
Experiment 3	97.74%	9.71%	99.61%	97.60%	05:34:30	01:48:12
Experiment 4	41.12%	7.48%	28.25%	93.78%	05:43:03	01:27:26
Experiment 5	41.54%	3.87%	27.85%	96.64%	05:37:01	02:17:16
Average	51.45%	11.80%	41.78%	93.09%	05:31:32	01:44:59

Table 4. Parallel RGSOM-CRV Experiments Result

	ACC	FAR	DTR/Recall	Precision	Training Time	Testing Time
Experiment 1	97.54%	11.02%	99.72%	97.26%	08:32:11	00:52:48
Experiment 2	94.79%	19.93%	99.48%	94.01%	06:15:59	00:42:02
Experiment 3	78.51%	17.52%	77.26%	93.34%	05:51:22	00:57:07
Experiment 4	91.21%	42.68%	99.69%	90.32%	06:18:46	00:47:38
Experiment 5	97.27%	12.72%	99.79%	96.87%	05:48:16	00:33:42
Average	91.86%	20.58%	95.32%	94.35%	06:33:18	00:46:39

Table 5. Parallel RGSO-CRV Result Each Protocol at Experiment 5

Protocol	ACC	FAR	DTR/Recall	Precision
icmp	99.74%	56.80%	100.00%	99.74%
tcp	93.39%	14.97%	99.44%	90.17%
udp	98.48%	0.52%	33.41%	49.60%
Total	97.27%	12.72%	99.79%	96.87%

For more detail insight we can study with the map generated in each epoch, which shown at Figure 5-7. From the visualization shown at Figure 5-7 there are new knowledge of information about the training process of Parallel RGSOM. Figure 5 at udp protocol shown that nodes which separate randomly is nodes that generated by reupdate unused reference vector process. The randomly separate of some node also appear at Figure 7 for TCP and udp protocol. At Icmp protocol from first until fourth epoch shown that decreasing of used reference vector number, that mean there are more similar weight in training data.

Parallel RGSOM-CRV is outperform regular RGSOM in efficiency of time consume, its spend average 6 hour and 33 minutes and 18 second while training, and 46 minutes and 39 second for testing. Training time consume when using RGSOM-CRV is better than Parallel RGSOM-CRV, this because in parallel RGSOM-CRV there are procedure to selecting input according to protocol type. However, time consuming for testing using Parallel RGSOM-CRV is better than RGSOM, this because at parallel RGSOM generate less used nodes in the map, so time for scanning the winner node more efficient. The problem of Regular GSOM for classified big data has been fixed by RGSOM-CRV and Parallel RGSOM-CRV, the restricted of nodes length generate by growing threshold make the limitation of map to growing bigger and bigger. Clustering reference vector make RGSOM-CRV capable to generalize weight base on growing threshold.

From Table 6, Parallel RGSOM-CRV has lower accuracy than other methods, but let see the number of testing data, proposed method have 4,898,431 instances as testing data, and have less feature to process. With a larger amount of tested data, Parallel RGSOM-CRV is capable of producing 91.86% accuracy, so this method very promising to solve the big data problems in classification.

Table 6. Comparison of proposed method with other methods by number of training data, tesing data, features, and accuracy.

Method	Training Data	Testing Data	Features	ACC %
KNN [1]	5,552	5,552	15	97.69%
SVM+ELM base K-mean [10]	494,021	311,029	41	95.75%
TAN+REP [12]	326,053	167,968	Not provided	98.99%
Parallel RGSOM-CRV	494,021	4,898,431	9	91.86%

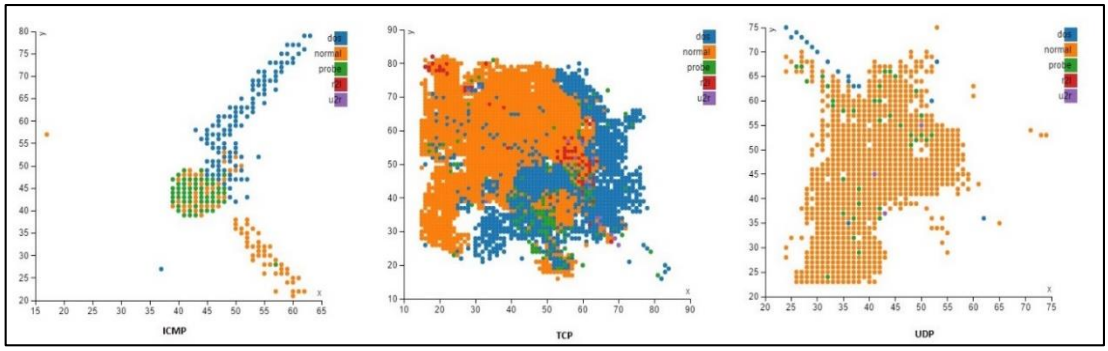

Figure 5. Map generated with Parallel RGSOM-CRV for first epoch at experiment 5

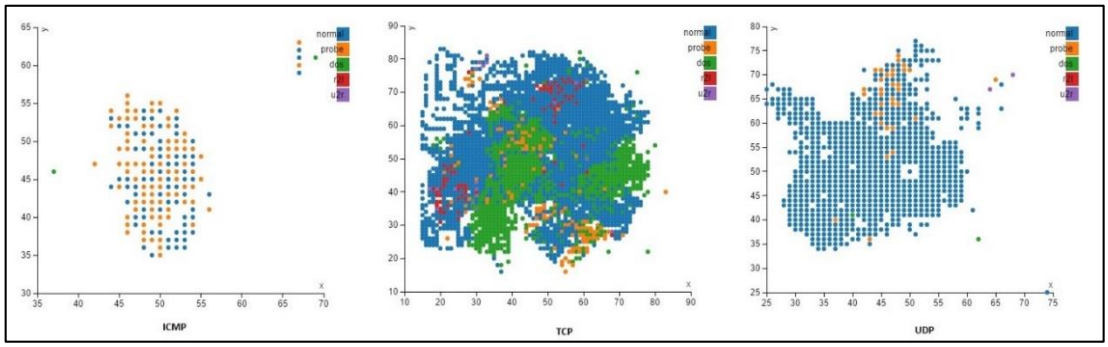

Figure 6. Map generated with Parallel RGSOM-CRV for second epoch at experiment 5

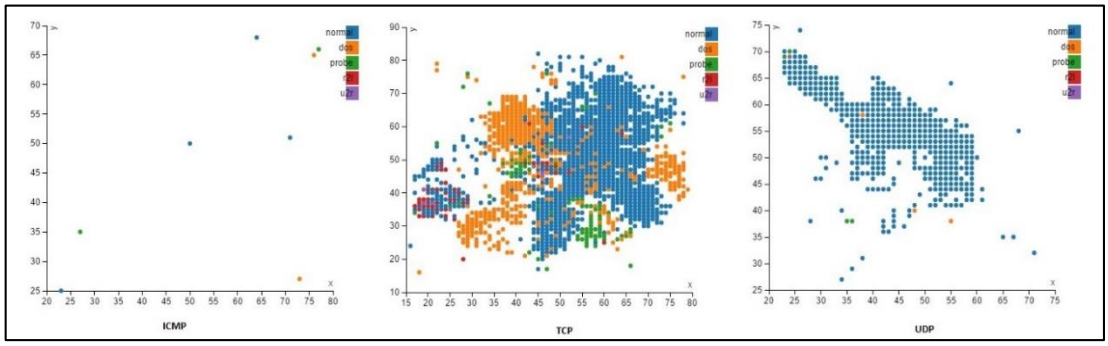

Figure 7. Map generated with Parallel RGSOM-CRV for third epoch at experiment 5

4. CONCLUSION

From this experiment Parallel and RGSOM-CRV outperform than regular GSOM in time consuming, so this propose method is more efficient than GSOM method, and from comparation with other method, the result of Parallel RGSOM is acceptable with 91.86% for accuracy, false alarm rate around 20.58%, recall or detection rate is 95.32%, and 94.35% in precision. This study also conclude that find the best of maximum node will increase the efficiency, and RGSOM generalize capability depend on GTstart and

GTStop. The capability to generalize reference vector make accuracy and detection rate acceptable. Finding the optimum parameter setting of growing thresholds can be used as a reference for the future research.

REFERENCES

[1] L. Li, H. Zhang, H. Peng, and Y. Yang. Nearest Neighbors based density peaks approach to intrusion detection. Chaos, Solitons & Fractals. 2018; 110: 33–40.
[2] E. Kabir, J. Hu, H. Wang, and G. Zhuo. A novel statistical technique for intrusion detection systems. Future Generation Computer Systems. 2018; 79: 303–318.
[3] M. Ahmed, A. Naser Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*. 2016; 60: 19–31.
[4] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez. Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems*. 2014; 71: 322–338.
[5] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida. A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection. *Computers & Security*. 2018; 75: 36–58.
[6] F. Al Huda, W. Firdaus Mahmudy, and H. Tolle. Android Malware Detection Using Backpropagation Neural Network. *Indonesian Journal of Electrical Engineering and Computer Science*. 2016; 4(1): 240-244.
[7] C. Xiang, Y. Xiao, P. Qu, and X. Qu. Network Intrusion Detection Based on PSO-SVM. T*ELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(2): 1502-1508.
[8] A. Karami and M. Guerrero-Zapata. A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing*. 2015; 149: 1253–1269.
[9] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença. Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic. Expert Systems with Applications. 2018; 92: 390–402.
[10] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*. 2017; 67: 296–303.
[11] G. Kim, S. Lee, and S. Kim. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*. 2014; 41(4): 1690–1700.
[12] K. Rajasekaran and K. Nirmala. A Novel and Advanced Data Mining Model based Hybrid Intrusion Detection Framework. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2015; 13(2): 223-231.
[13] J. Jabez and B. Muthukumar. Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. *Procedia Computer Science*. 2015; 48: 338–346.
[14] W.-C. Lin, S.-W. Ke, and C.-F. Tsai. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*. 2015; 78: 13–21.
[15] T. Kohonen. Essentials of the self-organizing map. *Neural Networks*. 2013; 37: 52–65.
[16] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*. 2000; 11(3): 601–614.
[17] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan. Cost-based Modeling and Evaluation for Data Mining with Application to Fraud and Intrusion Detection: Results from the JAM Project. 2000.