International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3609-3615

An Efficient DDoS Attack Detection Mechanism Based on Distributed Self Organizing Map

Yang-Ick Joo[1] and Minho Park[2*]

Division of Electrical and Electronics Engineering,
Korea Maritime and Ocean University[1]
School of Electronic Engineering, Soongsil University[2]

Abstract

Even in the era of 5G and beyond networks, we are struggling against an old problem, DoS (Denial of Service) attack which has arisen since the beginning of network. Since it is still the biggest threat of all cyber-attacks and it is growing more complex, there has been a lot of effort to mitigate DDoS attacks. As one of the various approaches, machine learning techniques are applied to detect attacks. Self-Organizing Map (SOM), one of unsupervised learning mechanisms, has been used as a tool to detect DoS attacks. However, existing SOM-based approaches have the potential drawbacks, the limited detection throughput and vulnerability to DoS attack. Therefore, this paper proposes a new concept of SOM, Distributed SOM (DSOM), and a novel DDoS attack detection mechanism based on DSOM. Through the simulation and experiment results, we show the proposed mechanism provides a feasible and efficient detection.

INTRODUCTION

DoS (Distributed Denial of Service) attack[1] , one of the most traditional and common attacks, still remains the biggest threat of all cyber-attacks, and will threaten the next generation networks persistently [9]. It tries to make network resources such as web servers be unavailable to intended users by monopolizing the resources. The principle of DoS attack is quite simple, which involves sending of IP packets to cause the saturation or instability of a victim. However, it is getting harder to detect these DoS attacks because they are evolving to be more intelligent and diverse. Moreover, anybody can easily perform DoS attacks even without any professional knowledge if they have attack tools. Therefore, it is still a major threat.

However, it is not straightforward to distinguish between normal and attack traffics because the headers of attack packets are forged as if they are normal packets. To detect the DoS attack, the comprehensive packet analysis is required, which may cause too much the packet analysis overhead.

For the packet detection, various data mining techniques have been proposed [1]. Among them, this work makes use of Self-Organizing Map (SOM) which is an unsupervised artificial neural network. For detection, SOM firstly is trained with statistical features of network traffic. After the first trained data, it reorganizes itself with the features of newly incoming traffic repeatedly, and classifies traffic based on itself. Because of its flexibility and adaptability to environmental changes, it has been used for detection [2-5].

However, SOM-based approaches have the potential drawbacks: the limited detection throughput and vulnerability to DoS attack. For example, consider a common large-sized network having multiple gateways to outside networks as shown in Figure 1.

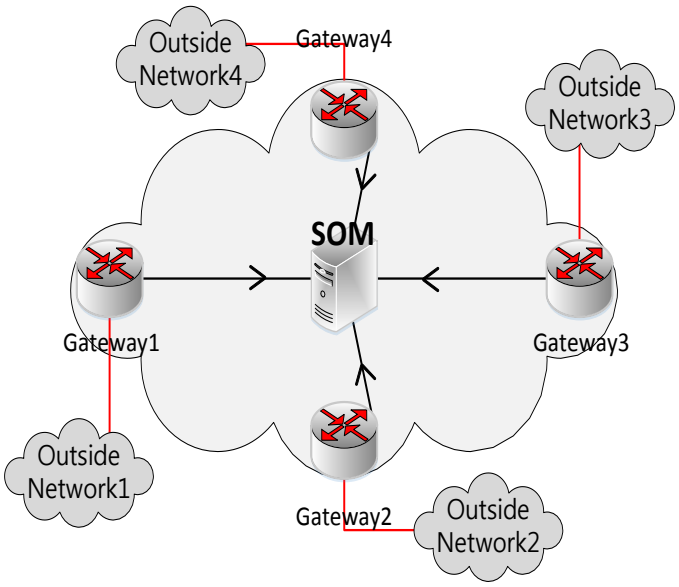


Figure 1: A common large-sized network

[1] *In this paper, Denial of Service is interpreted in a broad sense to involve both DoS and DDoS (Distributed DoS).*
**: Corresponding author*

To monitor all traffics coming through the multiple gateways and detect DoS attacks, each of the gateways has to send the information for reorganizing the SOM about its own traffic to a single point where the SOM is running. Due to the aggregation into the single point, the SOM can be the bottleneck of detection throughput or even a target of DoS attack. Ironically, a detection system for DoS attack prevention can be a victim for DoS attack. It is getting severer as the network size increases. This irony motivated us to investigate the vulnerability.

In order to resolve this problem, we came up with a Distributed SOM (DSOM) to make the existing SOM-based approach be more scalable and robust against DoS attack. The idea is to distribute the functions of an original SOM to multiple points. The multiple Distributed SOMs, called DSOM, are separately running at the corresponding points, e.g., the gateways to outside networks. Each DSOM monitors only its own traffic and detects attack traffic. Since there is no aggregation of traffic information into a SOM, we can avoid both the limitation of detectable traffic capacity and the vulnerability to DoS attack.

Two questions may arise because multiple DSOMs are separately reorganized, and running independently: 1) Do the distributed SOMs have the same detection rate? 2) Can we achieve the same level of detection rate as the original SOM? For example, if DSOM1 and DSOM2 have been trained and reorganized with different traffic, they have different maps. That means an attack packet might be detected by DSOM1, but not be detected by DSOM2. Therefore, we propose a weighted linear sum of multiple DSOMs to eliminate the deviation among them. Through comprehensive experiments with real traffic data, this paper shows DSOM is a feasible approach to achieve the reasonable detection rate. Also we implemented a test best with Raspberry Pie[10] to emulate gateway routers, and a web server, and evaluated the proposed DSOM performance.

The rest of this paper is organized as follows. In Section II, the background of SOM is described. In Section III, we introduce Distributed SOM. We verify the effectiveness of the proposed concept, the Distributed SOM in Section IV, and assess the performance of the proposed system through an experiment on the test bed in Section V. Finally, we make conclusions in Section VI.

SELF-ORGANIZING MAP

We can learn something without any teaching. For example, if we eat a food, we learn its shape, smell and taste. Afterwards, we know how it smells and tastes when we see the food. Like this process, learning without any external teaching is called an unsupervised learning.

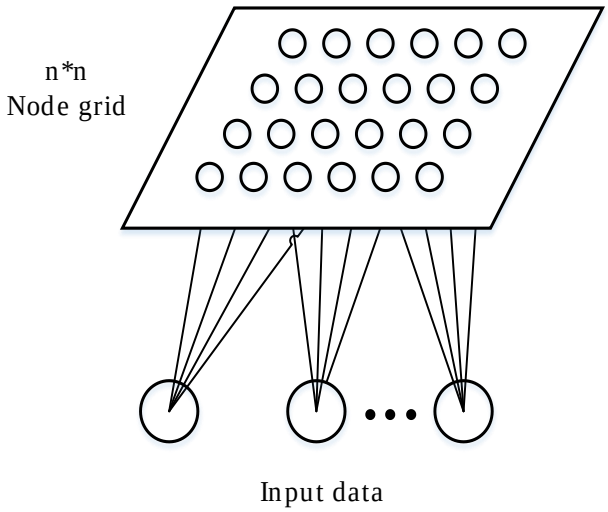


Figure 2: Self Organizng Map

Self-Organizing Map (SOM) is a data mining technique based on the unsupervised learning. As shown in Fig.2, input data is reorganized and mapped into a low-dimensional space called a map or a node grid. Since the input data is usually high dimensional, SOM is useful to visualize a low-dimensional views of high-dimensional data. The learning in SOM causes different parts of the network to respond to certain input patterns. In other words, the similar input patterns affect a nearby area of the map.

The learning process in SOM is described briefly. You can see the details of the SOM algorithm in [6].

Step 1) The vector for each node in the map is initialized with random or fix values.

Step 2) When an input vector is fed, its Euclidean distance to all nodes in the map is computed.

Step 3) The node whose distance is the closest to the input becomes Best Matching Unit (BMU).

Step 4) The radius of the neighborhood of the BMU is calculated, which diminishes at each time.

Step 5) Each neighboring node's vector is adjusted to make them more like the input vector according to the following equation:

$$W(t+1) = W(t)+L(t)*\Theta(t)*(V(t)-W(t)).$$

L(t) is the learning rate, which should decrease gradually over time. $\Theta(t)$ is the relative distance to BMU. The closer a node is to BMU, the more its vector gets affected.

Step 6) Repeat ***Step 2)*** over many iterations.

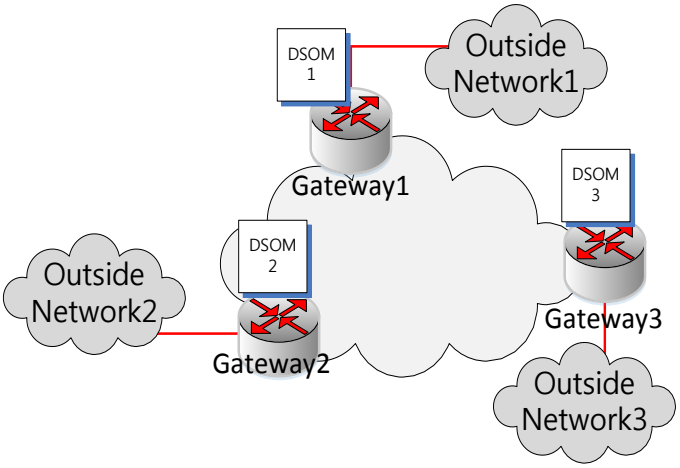


Figure 3: DSOMs are assumed to be running at different gateways.

DISTRIBUTED SOM FOR DOS DETECTION

In the existing SOM-based detection, every input data is aggregated into a single SOM. The SOM reorganizes itself with all the aggregated input data and classifies the type of the input data. Similarly, each DSOM is reorganized with its own input data. In Fig.3, DSOM1 at GW1 is reorganized with traffic incoming to GW1, and DSOM2 is also done in the same way. Since the input data affects how SOM is formed, DSOM1 and DSOM2 will be different. We name this difference Map Deviation. Due to this map deviation, SOM1 may classify traffic into normal even if SOM2 classified it into attack.

One more important consideration is Detection Accuracy. Since all input data is reflected to the single SOM in the existing SOM-based detection, the classification (detection) accuracy of the single SOM can be higher than that of distributed SOM which is reflecting a part of input data.

We hypothesized that two different small sized input data will not incur too much map deviation i.e., map deviation is small in a short time, and confirmed the hypothesis with real data, which will be described in the next section. However, it is natural that the map deviation increases as time goes. Therefore, we propose a weighted linear sum of multiple SOMs to eliminate the map deviation, which will be described in this section. Also, we found DSOM can achieve reasonable detection accuracy compared to the original SOM.

DSOM Overview :

Fig.4 shows the overview process of DSOM. The process consists of three steps.

Step 1) Initializing: Each DSOM is trained with a initial input data. There is no map deviation.

Step 2) Separate operation: With the initial map, each DSOM operates separately. On receiving its own input data, it reorganizes its map and classifies the input data, e.g., normal or attack. This separate operation can last until map deviation is acceptable. This paper leaves how long this operation can be to a future work. In this paper, we assume it is 1 minutes.

Step 3) Weighted sum of SOMs: If each DSOM continues to operate separately, the map deviation becomes so big that each DSOM can make a different result even for the same input. To remove the map deviation, DSOMs are periodically merged into a single SOM in a weighted sum manner as follow:

$$\sum_j \frac{amt_j}{\sum_i amt_i} \times DSOM_j \ .$$

Of course, an original SOM made from all the input data is not the same as the weighted sum. However, it can achieve reasonable detection accuracy. This weighted sum of SOM becomes an initial map of DSOMs, which is the initializing step.

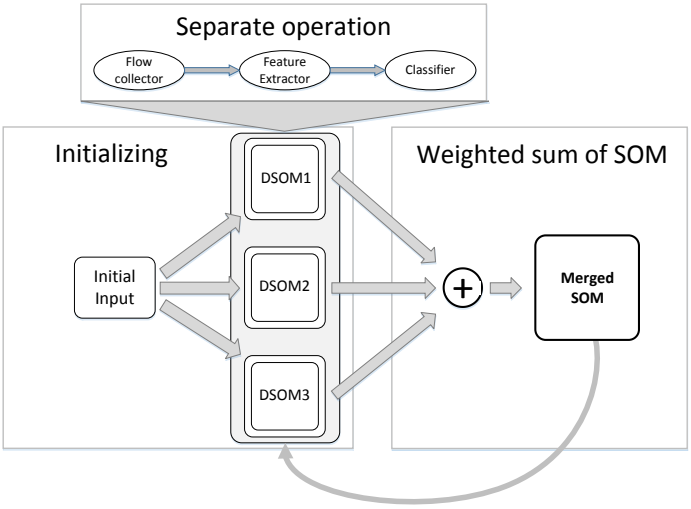


Figure 4: DSOM Overview

Inside DSOM :

Each DSOM agent [2] consists of the following three components.

1) Flow collector: It collects the input data for SOM. The input data in this DoS detection is the packet information.

[2] The DSOM agent is a program operating DSOM.

2) Feature Extractor: It extracts the features of traffic for every source IP. According to the traffic types, the features varies. It means different attacks have different features. Because the main goal of this work is to show the feasibility of DSOM, only two types of attacks, ICMP flooding and TCP Syn flooding, are considered. The following six tuples are extracted, which charaterizes every source IP.

   a) number of flows: One of the main features of DoS attacks is the number of flows. The characteristics of the ICMP and TCP syn flooding attacks are different. ICMP flooding has only a one flow and TCP syn flooding attack produces a higher number of flows.

   b) number of packets/flow: Another peculiarity of DoS is the packet size. ICMP flooding has a large number of packets per flow. TCP syn flooding has fewer packets, usually between 1 and 3 per flow.

   c) number of bytes/flow: Similarly, the number of bytes/flows is a good feature to classify the DoS attacks. While ICMP flooding attack has a high number of bytes/flow since the number of flows is very small, TCP syn flooding attack has small bytes/flow between 60 and 200 bytes per flow.

   d) growth of client ports: Under ICMP flooding attack, there is no change in the port of attackers. In the case of TCP syn flooding, the number of ports increases highly since attackers send syn packets via a number of different ports.

   e) duration: This indicates how long a client and a server have been connected. In ICMP flooding, attackers tend to connect to a server for a long time. However, TCP syn flooding has a short duration since it spoofs its IP and port continuously.

   f) Protocol: It is important which protocol is used for communication. Each protocol is represented as a specific integer, e.g., TCP 0 and ICMP 1.

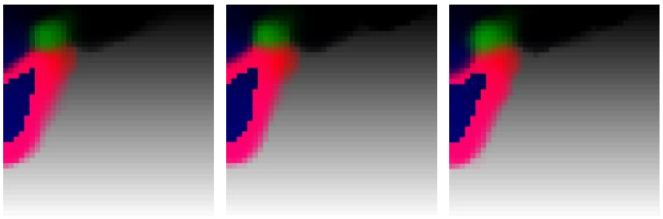
3) Classifier : It constructs input vectors from the extracted features. The classification process is the same as the original SOM. Our classifier introduced k-means clustering algorithm to determine what type of traffic the input data is. K-means patitions n observations into k clusters. This work partitioned each DSOM map into 20 clusters and the same type of traffic belongs to the same cluster. For example, ICMP flooding and TCP Syn flooding attack traffics belong to the cluster {2,4} and {11,12,13}, respectively.

EFFECTIVENESS VERIFICATION OF DSOM

Many previous works have shown the original SOM is a good enough technique to detect DoS attacks. Since the original SOM is changed to the distributed SOM in this work, we show that the distributed SOM works well as the original SOM does.

Map Deviation :

Since the input data affects SOM formation, separate DSOMs receiving different traffic should not be the same as each other, which is called Map Deviation. We consider a network shown in Fig.3, where three DSOMs are running at three different gateways. Obviously, the three DSOM are reorganized differently since the traffic coming to each gateway is different. However, our hypothesis is that the map deviation is negligible in a short time. To confirm the hypothesis, we conducted experiments with real traffic data [7-8]. Each DSOM is trained with the same initial input data for 3000 source IP addresses, and then it is reorganize with 3 different input set, each of which is 2000 source IP addresses. Finally, we tested how accurately each DSOM detects attackers with the same data. Fig.5(a) is the visualization for three DSOMs. Although they are slightly different from each other, the detection performances are almost same as shown in Fig.5(b). There are 504 normal source IP addresses and 496 attacker source IP addresses among 1000 test inputs. DSOM1&2 detected 492 of 496 attackers and DSOM3 detected 482 of 496 attackers. We, therefore, can consider the map deviation is ignorable.



DSOM1 DSOM2 DSOM3

(a) Visualization of DSOMs

		Test	detect	Accuracy
DSOM1	Normal	504	477	94.64%
	Attack	496	492	99.19%
DSOM2	Normal	504	477	94.64%
	Attack	496	492	99.19%
DSOM3	Normal	504	482	95.63%
	Attack	496	482	97.18%

(b) Test results

Figure 5: DSOM experiment results

Detection Accuracy :

Since the map deviation is ignorable only for a short time, we have to make DSOMs synchronized with each other to

guarantee that different DSOMs perform the same detection. We proposed a weighted sum of DSOMs and show the performance of the weighted sum SOM. We consider 3 different cases, Case1) an original SOM, Case2) three DSOMs with traffic ratio 1:1:1, Case3) three DSOMs with traffic ratio 1:2:3. The experiments have been conducted with 10000 inputs. In Case1, an original SOM was trained with 9000 of 10000 inputs. DSOMs are initialized with 3000 inputs. Then three DSOMs in Case2 were trained with three 2000 inputs, and three DSOMs in Case3 were trained with 1000, 2000, 3000 inputs, respectively. Then all the cases use the same 1000 inputs to compare the detection performances.

Table 1: Comparison of detection performance

	TP	TN	FP	FN
Original	95.7	96.77	4.3	3.23
1:1:1	99.19	94.44	0.81	5.56
1:2:3	100.0	92.86	0.0	7.14

Table1 shows the performance comparison. T, F, P and N denote True, False, Positive and Negative, respectively. TP (True Positive) means the probability that normal traffic is considered as normal. Likewise, FN (False Negative) means the probability that normal traffic is considered as attack. The original SOM has the slightly better performance in attacker detection. However, the proposed DSOM outperform the original SOM in normal traffic detection. Through these experiment results, it is shown that the proposed DSOM can detect DoS attacks at the almost same level as the original SOM.

PERFORMANCE EVALUATION

We implemented a test bed for the performance evaluation of the proposed DSOM as shown in Figure 6. The test bed consists of three raspberry pie boards [10] and a server. Each of the raspberry pie boards emulates a gateway router connecting to an outside network, and operates its own DSOM agent independently.

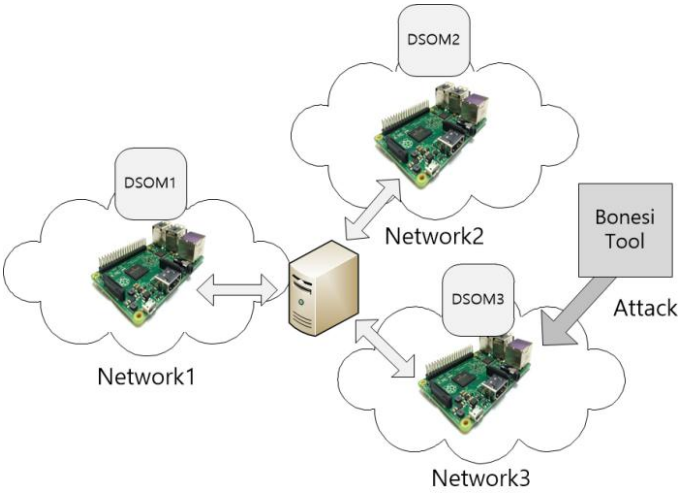


Figure 6: Test bed for experiment

Three components of the DSOM agent described in Section III.B are implemented in the raspberry pie board. The flow collector captures all incoming packets by using libpcap [11], and makes a pre-defined data structure periodically, which is the set of flows. We set the period to 1 minute in the experiment.

The feature extractor extracts characteristics from the flow set, and generates six-tuple sets composed of the number of flows, the number of packets/flow, the number of bytes/flow, the growth of client ports, duration, and protocol.

The classifier applies the six-tuple sets to its DSOM map, and reorganizes the map, which is the self-organizing process. Then it classifies each flow into the corresponding cluster of the map, by which the DSOM agent can distinguish which is a normal or attack flow. As described before, DSOM agents send their DSOM map to the server in order to avoid the map deviation.

The server collects multiple maps from the DSOM agents, makes a merged SOM through the proposed weighted sum, and sends it back to each DSOM agent. After receiving the merged map, each DSOM agent running on a gateway replaces the existing map with the new merged map. Then it resumes the DSOM operations.

The experiment used BoNeSi[12] which is a DDoS Botnet Simulator to simulate DDoS traffic in the test bed. The BoNeSi generates attack packets to Network3. In this work, we consider two types of attacks, TCP flooding attack and ICMP flooding attack. The server visualizes the merged map as well as three DSOM map sent by DSOM agents in real time so as to check how the system works.

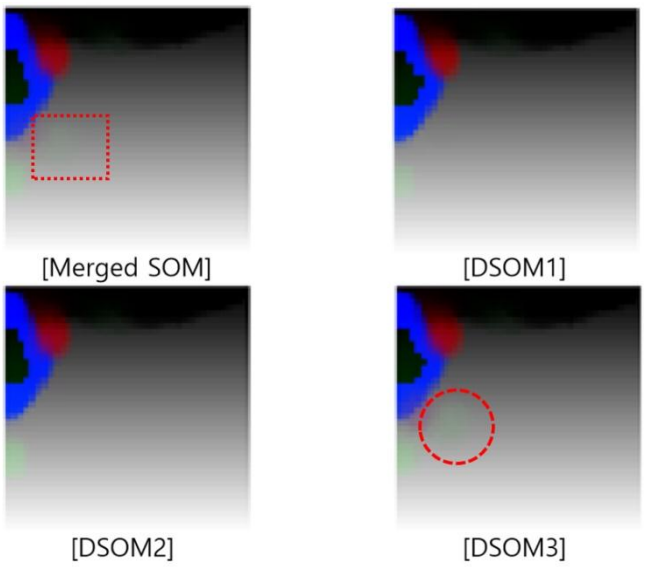


Figure 7: Map visualization after TCP flooding attack

Figure 7 shows the results after TCP flooding attack. Since gateway3 was attacked, the map of DSOM3 has different parts shown in the dotted circle. However, the attack traffic flows have been reflected to the merged map in the dotted rectangle, and the merged map is distributed to each DSOM agent again, which can overcome the map deviation. To assess the performance of the merged map, we compared the detection performance of DSOM1 agent before and after the weighted sum. While the DSOM1 agent detected only 13.33% of TCP flooding attack packets through the map of DSOM1, it detected 100% of attack packets after receiving the merged map as shown in Figure 9.

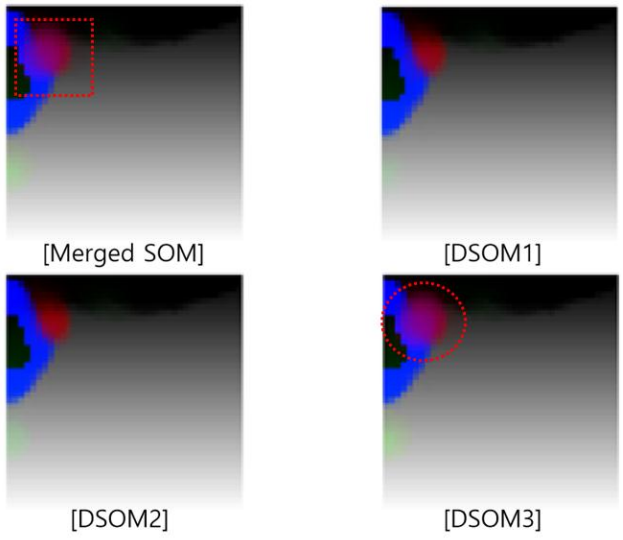


Figure 8: Map Visualization after ICMP flooding attack

Figure 8 shows the maps after ICMP flooding attack. Similar to the case of TCP attack, ICMP flooding packets changed the map of DSOM3 shown in the dotted circle. The merged SOM map is also changed shown in the dotted rectangle. We also compared the detection performance for ICMP flooding attack packets before and after the weighted sum. DSOM1 agent identified 43.33% of ICMP flooding attack packets. Although it is much higher than the rate of TCP flooding attack detection, DSOM1 agent with the merged map detected 100% of the attack packets shown in Figure 9. Therefore, the proposed DSOM mechanism can be considered as an efficient DDoS detection system.

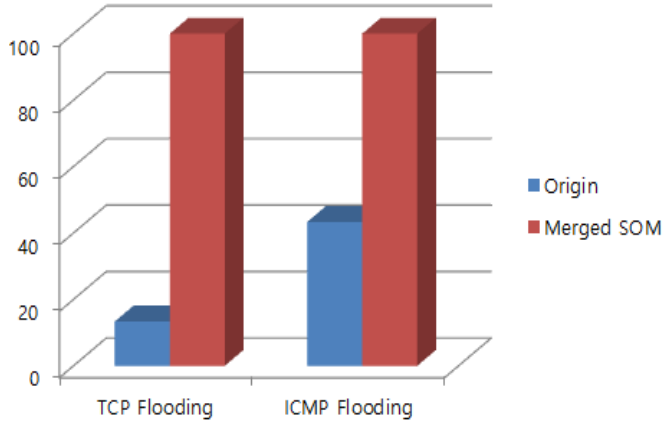


Figure 9: Detection rate comparison

CONCLUSION

This work proposed the Distributed Self-Organizing Map for DoS attack detection and showed the feasibility of DSOM. In the proposed approach, multiple DSOMs are running and detecting DoS attacks at different positions separately. Then they are periodically merged into a single SOM in a weighted sum manner to remove the map deviation. The experiments with real data and an attack tool showed DSOM is a feasible and efficient detection mechanism. In the future work, we will adapt this DSOM into Software Defined Network (SDN) environment.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2016R1D1A1B03933405)

Conflict of Interests

The authors declare no conflict of interests.

REFERENCES

[1] Chauhan, Amanpreet, Gaurav Mishra, and Gulshan Kumar. Survey on data mining techniques in intrusion detection. Lap Lambert Academic Publ, 2012.

[2] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalousnetwork traffic with self-organizing maps," in Recent Advances in Intrusion Detection. Springer, 2003, pp. 36–54.

[3] H. Gunes Kayacik, N. Zincir-Heywood et al., "A hierarchical SOMbased intrusion detection system," Engineering Applications of Artificial Intelligence, vol. 20, no. 4, pp. 439–451, 2007.

[4] M. Li and W. Dongliang, "Anormaly Intrusion Detection Based on SOM," in Proceedings of the 2009 WASE International Conference on Information Engineering-Volume 01. IEEE Computer Society, 2009, pp. 40–43.

[5] D. Jiang, Y. Yang, and M. Xia, "Research on Intrusion Detection Based on an Improved SOM Neural Network," in Proceedings of the 2009 Fifth International Conference on Information Assurance and Security- Volume 01. IEEE Computer Society, 2009, pp. 400–403.

[6] http://www.ai-junkie.com/ann/som/som1.html

[7] The CAIDA UCSD "DDoS Attack 2007" Dataset

[8] http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[9] The CAIDA UCSD Anonymized Internet Traces 2010 - [dates used],

[10] http://www.caida.org/data/passive/passive_2010_dataset.xml

[11] Global application and network security report, https://www.radware.com/ert-report-2015

[12] Raspberry Pie, https://www.raspberrypi.org/

[13] Libpcap, http://www.tcpdump.org/

[14] BoNeSi, [12] https://github.com/markus-go/bonesi