ISSN 2278-3091
Volume 8, No.1.4, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse0781.42019.pdf
https://doi.org/10.30534/ijatcse/2019/0781.42019

Nature Inspired Evolutionary Algorithm (ACO) for Efficient Detection of DDoS Attacks on Networks

D.Yuvaraj[1], M.Sivaram[2], A. Mohamed Uvaze Ahamed[0], S.Nageswari[3]
[0]Department of Computer Science, Cihan University - Erbil, Kurdistan Region, Iraq
Email:mohamed.sha33@gmail.com
[1]Department of Computer Science, Cihan University - Duhok, Kurdistan Region- Iraq
Email: yuva.r.d@gmail.com, yuvaraj.d@duhokcihan.edu.krd
[2] Department of Computer Networking, Lebanese French University – Erbil, Kurdistan Region, Iraq
Email: phdsiva@gmail.com,sivaram.murugan@lfu.edu.krd
[3]Department of Computer Science and Engineering, Bharath Niketan Engineeing College, Theni, India,
Email:saimonissh@gmail.com

ABSTRACT

Among the various attacks found extensively in the literature distributed denial of service attack is a special form of attack which poses to be a great menace and if not properly dealt with has the capability of bringing the power of computing systems to a halt with severe financial losses. Of the several defence mechanisms found in the literature, the most prevalent and prominently used ones are the intelligent and soft computing based evolutionary algorithms. Three such algorithms have been taken, investigated and experimented in this thesis for defence against DDoS attacks. This paper investigates the last algorithm namely ant colony optimization (ACO) which is yet another nature inspired algorithm for providing optimality in the DDoS defence system implemented. The last part of this chapter provides a comparative analysis of all the three implementations with respect to certain network critical parameters and inferences drawn based on the research findings.

Key words: Network attacks, distributed denial of service attacks, Ant colony optimization, error convergence.

1. INTRODUCTION

A general scheme of DDoS attack has presented in figure 1 where malicious traffic and attack on the victim system and network has been carried out. Distributed denial of service attacks have been seen to cause excessive depletion of resources and CPU clock and memory by flooding them with a continual and infinite loop of packets of information with scrap data causing a network and internet speed slow down [1, 2]. This phenomenon is comparable to people thronging the internet online shopping mart for a particular product which available in a limited stock. Another classical example could be the online reservation system where the network speed undergoes a drastic slowdown due to simultaneous multiple users flocking to get the same service to be tended to. Some terminologies with respect to DDoS attack detection [3] systems are presented below.

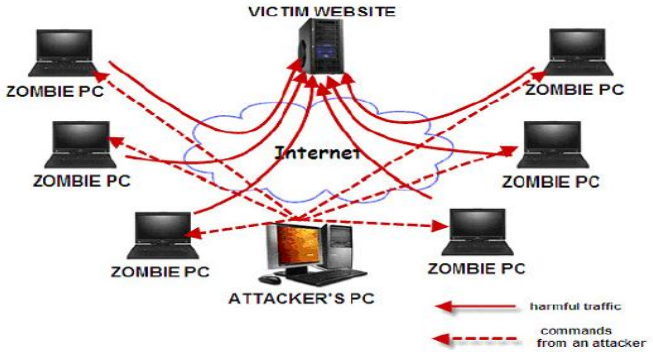


Figure 1: Scheme of attack of DDoS using Zombies

Bots are programs controlled by software architecture and management which execute a particular job based on the command issued to it from the control center [4]. However, bad bots causes malicious spreading and infection on other nodes in the environment and act as suitable agents for hackers to attack a particular system. Bots act as agents to the hacker and interact or engage directly with the system resource or the network connecting the two parties

Daily Statistics for August 2014 (Webalizer)

Day	Hits		Files		Pages		Visits		Sites		KBytes	
1	9929	1.46%	9319	1.40%	2166	0.50%	923	3.04%	1175	9.20%	114979	1.85%
2	7919	1.16%	7462	1.12%	1859	0.43%	841	2.77%	1392	10.90%	93971	1.51%
3	8160	1.20%	7876	1.18%	1448	0.34%	840	2.77%	569	4.46%	94925	1.52%
4	11730	1.72%	11175	1.68%	2118	0.49%	1116	3.68%	835	6.54%	138651	2.23%
20	10690	1.57%	10000	1.49%	1505	0.35%	1038	3.42%	787	6.16%	126669	2.03%
21	9147	1.34%	8853	1.33%	1287	0.30%						
22	60997	8.97%	60786	9.13%	53329	12.41%						
23	65660	9.65%	65347	9.82%	58729	13.66%						
24	62972	9.26%	62711	9.42%	56280	13.09%						
25	87124	12.81%	86744	13.04%	77463	18.02%						
26	34965	5.14%	34637	5.21%	27110	6.31%						
27	37570	5.52%	37187	5.59%	29586	6.88%						
28	35372	5.20%	35054	5.27%	26996	6.28%						
29	34833	5.12%	33789	5.08%	28057	6.53%						
30	26882	3.95%	26670	4.01%	20996	4.88%						
31	8348	1.23%	7853	1.18%	1738	0.40%						

One Bad Bot attacked our site 20,000 to 77,000 times per day for 9 straight days to bring about a DDoS Crash. IT FAILED.

Figure 2: Snapshot of statistics of Bot attack on a website

Bots as seen from previous section act as useful agents in the hands of attacker to introduce and infect malicious information and behaviour to nodes in the system. Botnets are primarily responsible for interconnecting these infected nodes together as a network to inflict maximum damage to the system and network. The organization structure of a Botnet is depicted in figure 3. Zombies are nodes or a physical machine connected to a network but has been compromised from its normal behaviour through infections introduced into it by an attacker.

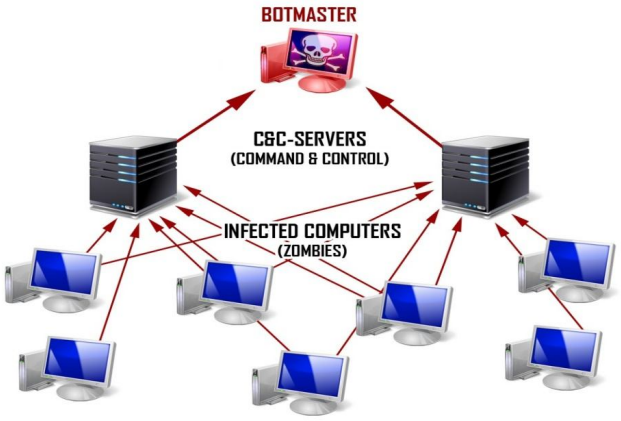


Figure 3: Organizational structure of Botnet

2. RELATED WORKS

Apart from the above techniques, multi-layer perceptron algorithms [5, 6] have also been reported in the literature which uses a data fusion algorithm to combine the various parameters in a hybrid approach. An advantage of MLP techniques is that it is self-adaptive and self-learning capable thus reducing the computation time and overhead. Experiments have been done for UDP flooding type of DDoS attacks and measurement of traffic taken as one important criterion for attack detection [7]. The proposed algorithm using MLP has been found to be effective in detection of attacks without disturbing the current data transfer protocol and parameters. The probability of false alarms has also been reported to be reduced in the literature using MLP.

Several optimization algorithms have also been discussed in the literature which includes the well-known ant colony optimization algorithm [8-11]. It is analogous in working principle to the natural phenomena of ants collecting their food. The behavioural pattern of ants for collection of food gatherings from one place to their destination is taken as the motivation behind the problem formulation for the optimization problem. Another optimization algorithm is the Bee's algorithm which also is a nature based optimization technique analogous to the behaviour of bee's to collect their food substances. This method utilizes less memory and offers a quick and optimized response. The principle is based on the initiation of the scout bee [11, 12] which goes on a scouting operation for food and the area for extraction of food is determined by the bee based on a suitable merit or wellness function. Wellness is defined as the specific employment with the available local resources.

Bin-packing issue includes the pressing of the objects of given size into containers of given limit. In one-dimensional Bin-packing the span of each protest is the genuine number in the vicinity of 0 and 1 and size of each container is same, given that the total of the quantity of items in the receptacle must not surpass 1. Bin-packing calculations utilized best fit calculations of assets in cloud. A formal meaning of BPP [13] can be characterized as a given rundown of items and their weights, container measure, locate minimal number of canisters so that every one of the assets are relegated to that receptacle. Other prominent techniques observed in the literature are the particle swarm optimization which is also based on a natural phenomenon [11]. The motivation behind the formulation of this algorithm lies in the swarming nature of particles around a point of high density from low concentrated adjoining regions. Neural network implementations [14, 15] have been extensively found in the literature due to their prominence and superior performance especially in cases of large data and high volume of network traffic. They are characterized by self-adapting and self-learning process which makes them an ideal tool for implementation of attack detection and defence mechanisms.

A Kohonen self-organizing structure [16, 17] is found in the literature which also provides reduction in data dimension utilizing a single feed forward with each node of the input connected to every other node in the output layer. The output is generalized to two outputs namely normal and malicious packets. Neurons in the implementation are organized into grids and the neuron with the best weight associated with it is termed as the winning neuron. Several variations of neural network implementations have been observed in the literature which utilized concepts of parallel and distributed computing in environments of huge number of massively connected components. Multilayer feed forward implementations have proved to be an effective solution in such cases where all neurons in the input layer are connected to all other neurons in the adjacent in a unidirectional progressive manner [18, 19]. This ensures that the broadcast of information can only be in a single direction from left to right. The learning has been accomplished by using back propagation neural networks for training the network for error convergence. Convergence to near zero levels have been observed in the literature with less number of iterations resulting in reduction of computation time.

Other implementations include the recurrent neural network known as Elman neural network in which every neuron in a particular layer receives input from every other neuron in the network. They do not follow a layered arrangement and experimental observations indicate a good convergence but at the cost of increased computational cost. A variant of neural network in cascade connection where each of the inputs, the

hidden layer nodes and outputs are connected through weights to the output is known as the cascade realization. The weights are adjustable and hence exhibit an adaptive behaviour. The error for the network is computed as usual initially only with one set of input and output units. After the first step, a hidden unit is added gradually and the weights are updated to minimize the error. The error defines the correlation between the obtained and the desired outputs. The weights are frozen at this moment and another set of hidden units are added and the process iteratively repeated. The repetitive procedure is stopped once the error comes down below a tolerable level.

3. PROPOSED WORK

3.1 Final Stage

Optimization algorithms are playing a major role in recent time as most of the optimization algorithms are intelligent enough and self-adaptive to dynamically changing input patterns. Further they follow a gradual and accurate convergence to minimal function defined by the problem objective and give precise readings of optimality. Off late, optimization algorithms have found great utilization network based problem objectives ranging from cloud and cluster computing, internet of things and bid data handling problems. Some of the optimization algorithms have been developed in inspiration of natural phenomena or observations of the behavioural patterns of natural creations. Examples could be particle swarm optimization algorithm which has been investigated in chapter four of this thesis, bee colony optimization algorithm deriving its motivation from the behaviour of bee's, fish swarm optimization motivated by the migration and food habit schemes of fish and groups of fish etc., One such naturally phenomena based algorithm investigated and experimented in this chapter is the Ant colony optimization (ACO) deriving the motivation from the food collecting behaviour of ants or colony of ants. The foraging behaviour of ants is the basic platform on which the ant colony optimization technique has been built.

Ants are well known to be social creatures living in colonies governed by certain principles and discipline inside the colony for the objective of survival. Procurement and storage of food by ants to keep them surviving during all seasons is the basic motivation behind the ACO. The ACO technique is quite similar to the conventional travelling salesman problem (TSP) whose primary objective is to find the shortest path between two communicating nodes. The process of ant colony behaviour is depicted in figure 4.

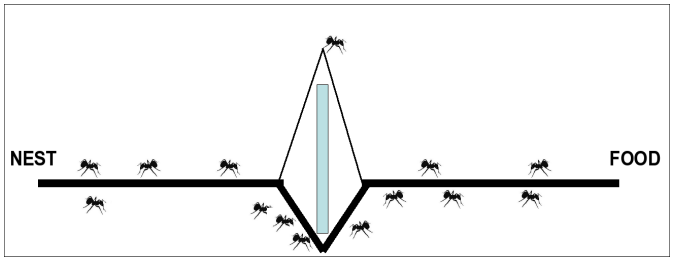


Figure 4: Ant colony foraging illustration

As seen from the above figure, the process of food foraging is initiated by a set of ants who scout the area or the surroundings for presence of food particles in a random fashion. Once the process is initiated, the ants leave a pheromone trail when they return back to their nest. In the second phase of this process, the ants smell the pheromone trail left by them and trace them back to the food source they had initially scouted. A stage by stage illustration of this process is depicted as a four stage process from figure 5 to 9.

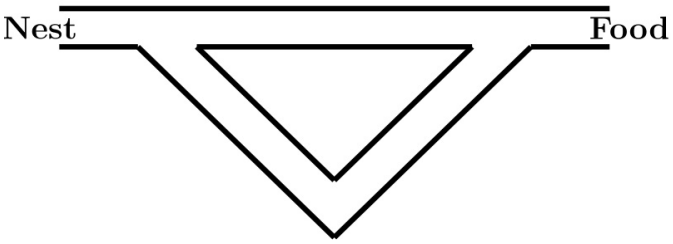


Figure 5: Phase I of ACO

In phase I, the ants remain inside the nest and no pheromone trail is left from the source to food.

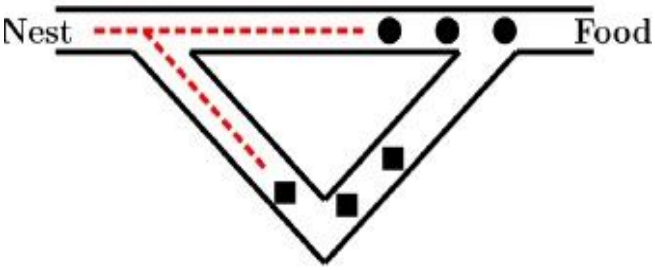


Figure 6: Phase II of ACO

In the second phase the ants initiate from process from the nest to the food source and leave a pheromone trail from the food to nest direction as depicted in figure 7. It could be seen from the above figure that the ants take up two paths the shortest and longest path from the food to nest.

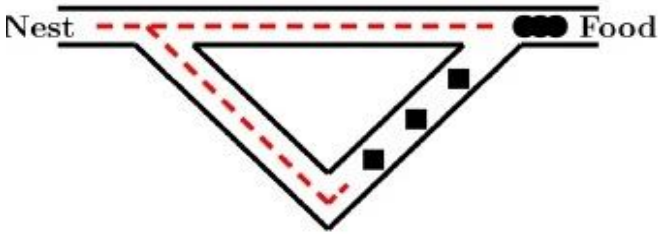


Figure 7: Phase III of ACO

In phase 3, the ants after having arrived at the nest from the food source from the two paths take subsequent trips to the food source based on the probability of the path travelled during the previous trip. The more the probability of ants taking the shortest path, the more the chance of them taking the shortest path in the next trips.

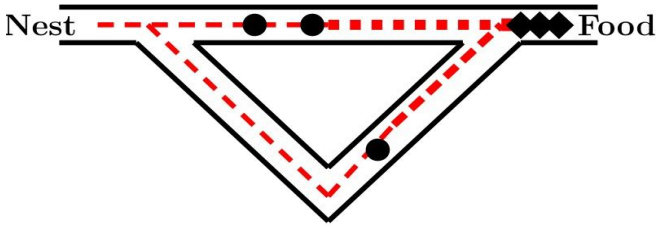


Figure 8: Phase IV of ACO

In the final phase it could be seen that the pheromone trail evaporates due to less probability of the ants taking up the least travelled path. The pheromone trails are indicated by the black circles in the above figure.

The following assumptions and inferences are considered in formulating the proposed ACO algorithm for DDoS attack detection and defence mechanism.

- The ants used in the proposed model travel in a perfectly synchronized and coordinated manner while natural ants travel in an asynchronous manner.
- As observed from the illustrations in previous sections, it could be seen that the natural ant behaviour invokes pheromone reinforcements based on the probability of shortest path being travelled more. Analogously a suitable quantifying parameter has been used to evaluate the reinforcement of pheromone on exhaustion in successive iterations.
- Natural phenomena is characterized by ants leaving pheromone trail while the ants in the proposed model deposit artificial pheromone during transit from food to nest.

This section of the chapter elaborates the proposed algorithm for improved ACO convergence for detection of DDoS attacks. DDoS attacks pose to be a major challenge in depleting the resources from the system as well as the online resource pool. Hence an efficient defence mechanism is required to address and countermeasure the DDoS attacks and restore the computational efficiency of the system under attack in the quickest time possible. The proposed model for implementation of ACO based DDoS defence system is shown in figure 9.

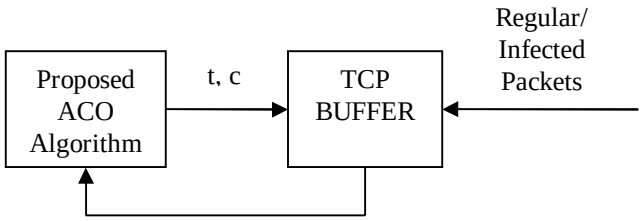


Figure 9: Proposed defence system architecture based on ACO

The standard procedure incorporates the packet size and frequency of event, retrenchment coefficient parameter, pursues and prey conduct. In the prey conduct, the distinct free development is considered from one spot to another spot. If ACO of l is moved to new location $\vec{l}_t$ the following actions must be takes place.

In the event that the new position is distinguished dependent on random distribution, at that point a perception procedure begins for the solidness.

$$\overrightarrow{l_{j,t}} = \vec{l}_{i,t} + Real * Randint \quad (1)$$

On the off chance that the stability of position is superior to anything the present position and the new position is refreshed

$$\overrightarrow{l_{j,t}}(n+1) = \vec{l}_{i,t}(n) + \left(\overrightarrow{l_{j,t}}(n) - \vec{l}_{i,t}(n)\right) * Randir \quad (2)$$

The above steps are rehashed many numbers of times and it is as often as possible refreshed. ACO changed as an expanded rate as far as a separation between the present position and new position for each removal. In the event that the better spot is not distinguished subsequent to playing out the above step, it move with an arbitrary step. In all conceivable emphasis, the swarm changes its place from past to current position. To pursue the conduct of AF forward movement for one stage is given as;

$$\overrightarrow{l_{j,t}}(n+1) = \vec{l}_{i,t}(n) + \left(\frac{\overrightarrow{l_{j,t}}(n) - \vec{l}_{i,t}(n)}{\varphi_{i,t}}\right) * Randint[0,1] \quad (3)$$

Position of AF i follows the best. With the goal that the ACOi moves randomly and this pursue conduct outcomes in expanded intermingling rate. The collaboration behaviour

between them is done comprehensively. The Euclidean distance is determined among every one of the ants for detecting neighbors remove. For powerful improvement in convergence rate, the likelihood of diminishing the computational load must be done in all cases. The center which is gotten already is determined dependent on the intra-cluster. The function is least and it is given as;

$$\emptyset(z_1, z_2, z_3, \ldots z_n) = \sum_{i=1}^{n}\left(\iint_{x^3}^{\alpha}\|Z_i - l\| * \varphi(i)/2\right) \quad (4)$$

where the new cost facility is basic for distinguishing proof of least necessities with acceleration constants. The general an incentive for speeding up steady is not more than 4 and the representation is appeared in figure 10.

The Execution of the algorithm as given below:

1. Initialize the ant populace population and create variable individual random values dependent on parameters like advance and development of fish dependent on proceeds with trails.
2. Compare the stability factor and quality of people and the optimal state is obtained.
3. By choosing the better procedure strategy for swarm and follow process dependent on the state obtained in step 2.
4. For each procedure the area is keep on expanded or diminished until the following area is changed.
5. When the new area is distinguished and it is compared with the present location and once the optimal focuses are crossed at that point go to the new spot or else step 3 is repeated.

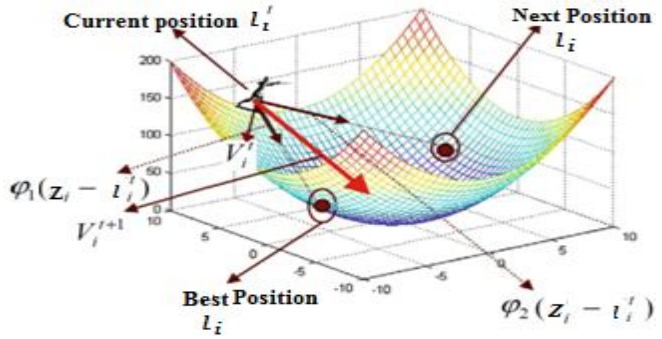


Figure 10: Illustration model of ACO minimization

The parameters are $\rho_2, \rho_3, \ldots \rho_n$

for $L_i = Randint(\rho_n L)$

Select the random points $u^t, t = 1,2, \ldots . L_i$

for $t = 1\ to L_i$

the set $z_n = u^t, Z_m = z(p^i, q^j)$

by computing the swarm with integer values

$$L_{swarm} = int(\rho_4 L)$$

$$if L_{swarm} > 0\ then$$

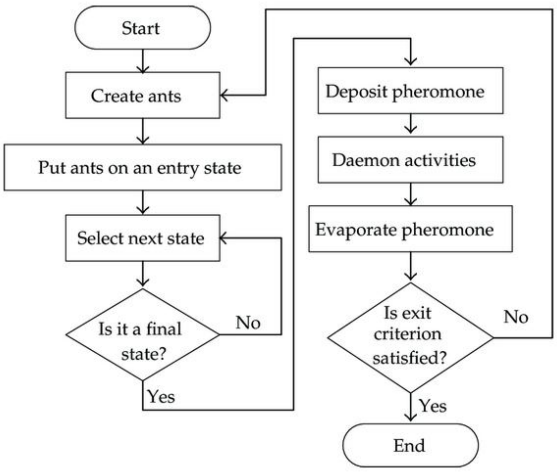
$$for i = 1\ to L_{swarm}$$

Perform the swarm move and create a suitable connection

the set $Z_\alpha = z(p_\alpha^i, q_\alpha^j)$

(5)

if $Z_\alpha < Z_m then$ replace the relating position into new once from the function value. The flow process of proposed



algorithm is depicted in figure 11.

Figure 11: Flow process of proposed ACO

4. EXPERIMENTAL ANALYSIS AND FINDINGS

The proposed framework is tried on a Celeron processor 1.85 GHz with 2GB RAM running Windows XP and coded by Matlab 6.5.The KDD Cup99 dataset is derived from the DARPA98 network traffic dataset by amassing singular TCP packets into TCP connections has been utilized for benchmarking the proposed ACO model. Every TCP connection has 60 features with a name which indicates the status of a connection as either being typical or a particular attack type.

Seven features namely *src_bytes, dst_bytes, count, srv_count, dst_host_count, dst_host_srv_count, dst_host_ same_src_ port_rate* have been chosen for the experimentation. The dataset contains about about two million connection records as test data and five million connection records as training data. 952 records have been chosen to limited data size.

Table 1: Comparative analysis of feature extraction

Input attacks	**Extracted Features**		
	ICR	**PT**	**ErF**
Itr800	86.89	1.921	10^{-9}
Itr720	59.28	1.918	10^{-8}
Itr 500	34.87	1.918	10^{-6}
Itr 340	30.25	1.918	10^{-6}
Itr 278	2.68	1.920	10^{-4}
Itr 164	1.99	1.917	10^{-4}

Itr 39	1.82	1.918	10^{-4}
Itr 20	0.47	1.920	10^{-2}
Itr 13	0.23	1.920	10^{-2}
Itr 2	0.25	1.920	10^{-1}
Itr 1	0.27	1.920	10^{-1}

The above table illustrates the measured values of features for varying degree of attacks simulated in the proposed algorithm. The attacks have been purely DDoS attacks only targeted over a period of time varying from 150 to 6000s on the target or victim server. The error convergence plot is depicted in figure 12.

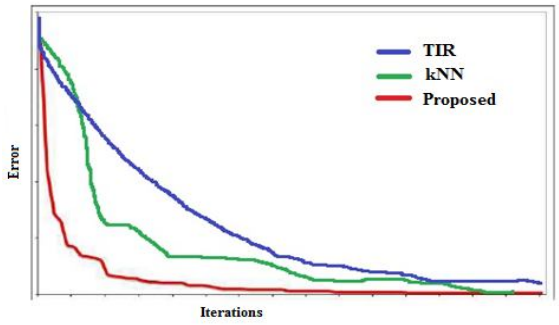


Figure 12: Plot of error convergence using ACO

The results have been observed for varying population number of 20 and 100 analysing the changes in the wellness of the worldwide best of the swarm at every cycle. It was seen that all the three populaces meet to the proportionate ideal arrangement, be that as it may, the little one requires a higher number of iterations (350) than the others, which need 339 and 347 iterations, exclusively. In addition, it tends to be seen how, by expanding the number of populations of ants, its decent variety and exploratory capacity remain essentially higher for a more drawn out number of iterations.

Simulations were carried out with ACO, ANN and PSO. The parameters used for analysis were the time taken to complete, accuracy of results and the scalability of the algorithms with respect to the size of the data. Data was passed to the algorithms as 5000, 10000, 20000, 50000 and 100000 records. Since network traffic is under consideration, the rules for attack detection tends to get higher, hence the size of data being used also plays a vital role in selecting an algorithm. The process was carried out in a system with2.5 GHz Intel Core i7 and 16 GB 1600 MHz DDR3 RAM.

Figure 13: Time Comparison plot

Figure 13 shows the time comparison between ACO, PSO and ANN. ACO, due to its heavy memory requirements was able to complete the set with 5000 records only. PSO and ANN performed well in all the five datasets. In terms of time ANN provides the best results, while the time taken by PSO increases with respect to the size of the dataset.

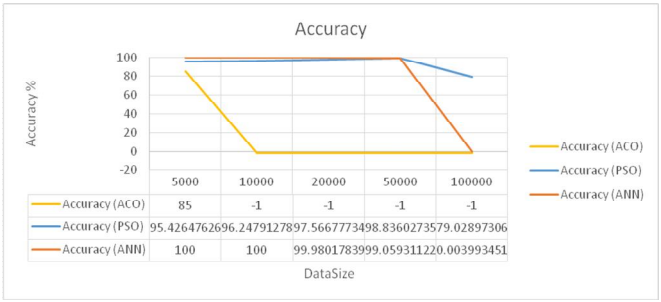


Figure 14: Accuracy Comparison plot

Figure 14 shows the accuracy comparison between ACO, PSO and ANN. ACO provided 85% accuracy, but had to be disregarded for other datasets due to its huge memory constraints. In terms of accuracy, it can be observed that though ANN performed well upto a dataset with 50000 records, another huge increase led to a steep drop in its accuracy. PSO, even though it shows some deflections, is comparatively stable towards data scalability.

5. CONCLUSION

This research paper has clearly elaborated the various features of the DDoS attacks which are quite essential in designing and implementing an efficient detection and defence system. Based on the literature findings of research contributions in chapter 2, the essential parameters namely time and connection have been used as training inputs to the optimization algorithm. The proposed algorithm has been compared with existing NN algorithm and the experimental results indicate marginally superior performance of the proposed position based ACO algorithm. The proposed algorithm has been modified with respect to the topology of implementation. The optimization capacity of this algorithm was improved in the high-dimensional space. Regardless, without sans scale systems are repetitive for perfect topology. The optimal solution of the entire gathering to complete the

speed is revived by following their very own optimal solution and other particles' optimal solution with a comparative measurement. This prompts an improvement in PSO algorithm which was additionally improved in taking care of the high-dimensional streamlining issue. Be that as it may, network topology optimization is not considered in the algorithm. The neighbourhood network topology of PSO impacts looking through the last optimal solution. The disadvantages of molecule swarm improvement (PSO) algorithm are that it is definitely not hard to fall into close-by ideal in high-dimensional space and has a low assembly rate in the iterative procedure. An uncommon number of examinations have been done to improve the PSO algorithm in the latest decades.

REFERENCES

1. Tariq Ahamad and Abdullah Aljumah (2014), "Hybrid Approach using intrusion Detection System", International Journal of Computer Networks and Communications Security, VOL. 2, NO. 2, pp. 87–92.
2. Salama, M. A., Eid, H. F., Ramadan, R. A., Darwish, A., & Hassanien, A. E. (2011). Hybrid intelligent intrusion detection scheme. In Soft computing in industrial applications (pp. 293-303). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20505-7_26
3. Vollmer, T., Alves-Foss, J., & Manic, M. (2011, April). Autonomous rule creation for intrusion detection. In 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS) (pp. 1-8). IEEE. https://doi.org/10.1109/CICYBS.2011.5949394
4. Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB journal, 16(4), 507-521. https://doi.org/10.1007/s00778-006-0002-5
5. Sarasamma, S. T., Zhu, Q. A., & Huff, J. (2005). Hierarchical Kohonenen net for anomaly detection in network security. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(2), 302-312. https://doi.org/10.1109/TSMCB.2005.843274
6. Ahamed, A. M. U., Eswaran, C., & Kannan, R. (2017). CBIR system based on prediction errors. J. Inf. Sci. Eng, 33(2), 347-365.
7. Kumar, P. A. R., & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. Computer Communications, 34(11), 1328-1341. https://doi.org/10.1016/j.comcom.2011.01.012
8. Teng, W. G., Chang, C. Y., & Chen, M. S. (2005). Integrating web caching and web prefetching in client-side proxies. IEEE Transactions on Parallel and Distributed Systems, 16(5), 444-455. https://doi.org/10.1109/TPDS.2005.56
9. Muthukumar G, George Dharma Prakash Raj (2014), "A Comparative Analysis on Symmetric Key Encryption Algorithms", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3, Issue 2.
10. Lai, G. H., Chen, C. M., Jeng, B. C., & Chao, W. (2008). Ant-based IP traceback. Expert Systems with Applications, 34(4), 3071-3080. https://doi.org/10.1016/j.eswa.2007.06.034
11. Ahamed, A. M. U., Eswaran, C., & Kannan, R. (2017). Predictive medical image compression using neural networks with gravitational search and particle swarm algorithms. In 20th International Workshop on Advanced Image Technology, Penang, Malaysia.
12. Lee, K., Kim, J., Kwon, K. H., Han, Y., & Kim, S. (2008). DDoS attack detection method using cluster analysis. Expert systems with applications, 34(3), 1659-1665. https://doi.org/10.1016/j.eswa.2007.01.040
13. Krömer, P., Platoš, J., Snášel, V., & Abraham, A. (2011, October). Fuzzy classification by evolutionary algorithms. In 2011 IEEE International Conference on Systems, Man, and Cybernetics (pp. 313-318). IEEE. https://doi.org/10.1109/ICSMC.2011.6083684
14. Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. IEEE transactions on computers, 65(10), 2986-2998. https://doi.org/10.1109/TC.2016.2519914
15. Sivakumar, V., Yoganandh, T., & Das, R. M. (2012). Preventing Network From Intrusive Attack Using Artificial Neural Networks. International Journal of Engineering Research and Applications (IJERA), 2(2), 370-373.
16. Ayoobkhan, M. U. A., Chikkannan, E., & Ramakrishnan, K. (2018). Feed-forward neural network-based predictive image coding for medical image compression. Arabian Journal for Science and Engineering, 43(8), 4239-4247. https://doi.org/10.1007/s13369-017-2837-z
17. Cirani, S., Ferrari, G., & Veltri, L. (2013). Enforcing security mechanisms in the IP-based internet of things: An algorithmic overview. Algorithms, 6(2), 197-226. https://doi.org/10.3390/a6020197
18. O'Shaughnessy, S., & Gray, G. (2011). Development and evaluation of a dataset generator tool for generating synthetic log files containing computer attack signatures. International Journal of Ambient Computing and Intelligence (IJACI), 3(2), 64-76. https://doi.org/10.4018/jaci.2011040105
19. Ahamed, A., Eswaran, C., & Kannan, R. (2018). Lossy image compression based on vector quantization using artificial bee colony and genetic algorithms. Advanced Science Letters, 24(2), 1134-1137. https://doi.org/10.1166/asl.2018.10702