Journal of Information Systems and Telecommunication
Vol.10, No.4, October-December 2022, 270-278
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Detection of Attacks and Anomalies in the Internet of Things System using Neural Networks Based on Training with PSO Algorithms, Fuzzy PSO, Comparative PSO and Mutative PSO

Mohammad Nazarpour[1], Navid Nezafati[2*] , Sajjad Shokouhyar[2]

[1]. Department of Information Technology Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[2]. Department of Management, Shahid Beheshti University, Tehran, Iran

Received: 13 Jun 2021/ Revised: 04 Nov 2021/ Accepted: 13 Dec 2021

Abstract

Integration and diversity of IOT terminals and their applicable programs make them more vulnerable to many intrusive attacks. Thus, designing an intrusion detection model that ensures the security, integrity, and reliability of IOT is vital. Traditional intrusion detection technology has the disadvantages of low detection rates and weak scalability that cannot adapt to the complicated and changing environment of the Internet of Things. Hence, one of the most widely used traditional methods is the use of neural networks and also the use of evolutionary optimization algorithms to train neural networks can be an efficient and interesting method. Therefore, in this paper, we use the PSO algorithm to train the neural network and detect attacks and abnormalities of the IOT system. Although the PSO algorithm has many benefits, in some cases it may reduce population diversity, resulting in early convergence. Therefore,in order to solve this problem, we use the modified PSO algorithm with a new mutation operator, fuzzy systems and comparative equations. The proposed method was tested with CUP-KDD data set. The simulation results of the proposed model of this article show better performance and 99% detection accuracy in detecting different malicious attacks, such as DOS, R2L, U2R, and PROB.

Keywords: Attack Detection; Internet of Things (IOT); Neural Network; PSO Algorithm; Fuzzy Rule; Adaptive Formulation.

1-Introduction

With the advancement of information technology, IT-related issues have also developed rapidly. The Internet of Things is a new model that integrates the Internet and physical objects belonging to different fields such as home automation, industrial process, human health and environmental monitoring. Having Internet-connected devices deepens our day-to-day operations, in addition to having many benefits, brings withmany security challenges. For more than two decades, intrusion detection systems have been an important tool for protecting networks and information systems. However, it is difficult to apply the former IDS techniques to the Internet of Things because of its special features such as limited resources, special protocol stacks, and certain standards. The proliferation of IOT has led to new challenges such as increased power consumption, more complex management due to increased data volume, more bandwidth demands to transmit IOT data, and use more powerful processors for information analysis. Moreover, protecting the privacy of individuals by protecting and safeguarding the information of individuals is very important and vital to achieve the commercialization of this industry [1-3]. Today, of course, the use of technologies such as optical fibers in the transmission of information and optical integrated circuits with Nano dimensions in fast processing and reducing energy consumption has greatly contributed to the commercialization of the Internet of Things. In contrast, the use of cloud storage, computing for data storage, processor and the use of SDN-based software pose a serious threat to attackers of the IOT infrastructure. Threats and anomalies created in the Internet of Things can be divided into four general categories: Dos attacks, R2L attacks, U2R attacks and Probing attacks. In Dos attacks, a large number of requests are sent to a system to disable it. In the U2R attacks, the intruder enters as the system administrator and destroys the system radically. In the R2L type of attack, the attacker enters the system as a local user and then takes control of the system by designing attacks. In the Probing attack, the intruder tries

✉ **Navid Nezafati**
n_nezafati@sbu.ac.ir

to obtain information from the system such as passwords, user numbers, important files and types of system services. One of the most important and popular tools in the field of attack prevention is the use of machine learning systems [4-5]. In this system, the system is modeled using artificial intelligence and based on existing experiences to prepare for predicting new conditions. Therefore, the system should be trained using training data that is the result of past experiences. One of the most powerful and efficient modeling tools in the field of machine learning is the use of artificial neural networks [6-7]. In simpler terms, neural networks are modern systems and computational methods for machine learning, knowledge display, and finally the application of knowledge gained to maximize the output responses of complex systems. The main idea of such networks is partly inspired by the way the biological neural system works to process data and information to learn and create knowledge. The key element of this idea is to create new structures for the information processing system. The system is made up of a large number of extremely interconnected processing elements called neurons that work together to solve a problem and transmit information through synapses (electromagnetic communications). In these networks, if one cell is damaged, other cells can make up for its absence and contribute to its regeneration. These networks are able to learn. For example, by injecting tactile nerve cells, the cells learn not to go to the hot body, and with this algorithm, the system learns to correct its error. Learning in these systems is adaptive, that is, using examples, the weight of the synapses changes in such a way that the system produces the correct response if new inputs are given. The main philosophy of the artificial neural network is to model the processing properties of the human brain to approximate conventional computational methods with the biological processing method. In other words,the artificial neural network is a method that learns the knowledge of thecommunication between several data sets through training and stores it to use in similar cases. This processor works in two ways similar to the human brain: Neural network learning is done through education. Weighting similar to the information storage system takes place in the neural network of the human brain.

An artificial neural network consists of three layers: input, output and processing. Each layer contains a group of nerve cells (neurons) that normally communicate with all the neurons in the other layers unless the user restricts communication between neurons; but the neurons in each layer have no connection with other neurons in the same layer. A neuron is the smallest unit of information processing that forms the basis of the function of neural networks. A neural network is a collection of neurons that, being located in different layers, form a special architecture based on the connections between neurons in different layers. Neurons can be a nonlinear mathematical function, so a neural network made up of a community of these neurons and can also be a completely complex, nonlinear system. In the neural network, each neuron operates independently, and the overall behavior of the network is the result of the behavior of multiple neurons. In other words, neurons correct each other in a process of cooperation. Figure 1 shows an artificial neural network versusthe neural network of the human body. The system inputs, called X1, X2, Xn, enter in the input neurons and transfer to the hidden layers via W1, W2…Wn. This transfer is done by multiple inputs on the W coefficients. Now apply the nonlinear function to the output layer to enhance the modelling application for nonlinear samples and data collections. In the learning procedure, the w coefficient is determined by machine learning algorithms. Now we want to focus on the weight coefficient determination. One of the interesting and attractive methods is the backpropagation algorithm for determination the W coefficients [8-9]. This method is based on the slope of error and has a good speed to response determination. Instead, it is trapped in the local optimum point and unable to find the global optimum [10-11]. One solution is to use meta-heuristic algorithms. A metaheuristic optimization algorithm is an innovative method that can be applied to various optimization problems with minimal modifications. Metamorphism algorithms significantly increase the ability to find high-quality solutions to difficult optimization problems.

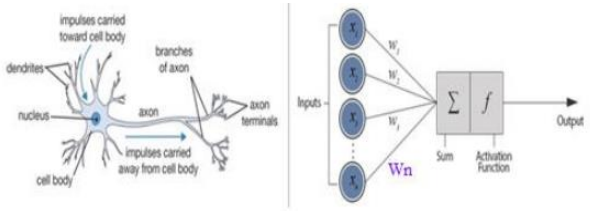

Fig. 1. schematic biological neuron (Left) versus artificial neural network (Right)

One of the evolutionary optimization algorithms that have a good performance speed is the PSO algorithm. Additionally, this algorithm, like other evolutionary algorithms (genetics and colonial competition and so on) has simpler calculations. In this article, we used the PSO algorithm to train the neural network. Then, we will show that although training by the PSO algorithm gives a much more accurate answer than the BP training method, it is still possible to reach much more accurate answers by changing the PSO algorithm. For this purpose, we used a combination of fuzzy, comparative and mutation methods to alter this algorithm and showed that we get very acceptable results by training the neural network by the altered PSO algorithm.

2-Methodology

This research work aims to propose a Neural Network Model of KDD-Data set for intrusion detection in IOTdevices. This part of the paper describes the proposed work methodology, i.e., proposed attack detection framework, proposed network model, data set description, and preprocessing.

a. The Framework of the attack detection

The proposed procedure is illustrated in Figure 2. As you can see in this flowchart, the data must first be collected for training the neural network. To collect the data, we use the kdd-cup data set. In the continuation of the data preprocessing operation is done, it includes deleting similar data, extracting more effective data, and normalizing the data. We then classify the data into two categories: training data and test data, so that test data makes up 20% of the data and training data makes up 80% of the data. In the next step we architected the neural network and trained it basedon PSO and Modified PSO algorithms and training data. Finally, the evaluation of the model created by the neural network is performed based on test data.

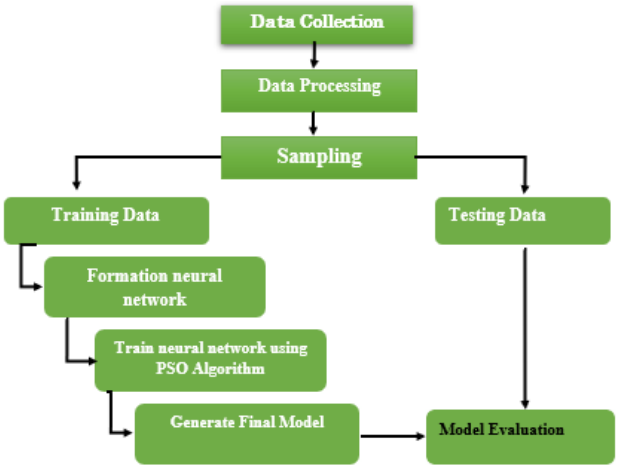

Fig. 2. overall framework of the attack detection using neural-network based PSO algorithm

b. Neural Network

An artificial neural network, also called a simulated neural network or neural network, is an interconnected group of artificial neurons that uses a mathematical or computational model to process information and based on the connection approach. One of the classic types of theartificial neural networks is the perceptron network. The following figure shows a perceptron neural network:

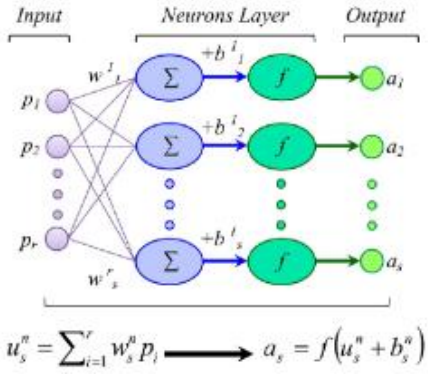

Fig. 3. Multi-layer Perceptron neural network.

A multilayered (deep) perceptron neural network will result from the stacking of several perceptron's. That is, we will have multiple layers of neurons in such a network. Here we have an output layer and an input layer. There are also several layers of neurons between the input and output layers. The layers between the input and output layers are called the Hidden Layer. Layers that are close to the input layer are usually called bottom layers. Layers that are close to the output layer are also called top layers. Except for the output, each layer has a bias. A network that has a large number of hidden layers is called a Deep Neural Network. As mentioned, there are several classical methods for determining weight and bias coefficients. But all of these methods are caught in local optimal points and are not able to determine the global optimal point. To solve this problem, in this paper, we use the training method based on thePSO optimization algorithm and extract these coefficients for system modeling. In addition, despite the high speed of the bird algorithm, it does not have enough accuracy and to improve the system, we use fuzzy, mutation and adaptive models to increase the accuracy of neural network performance in addition to speed.

c. Classical PSO

Particle swarm optimization algorithms are one of theheuristic optimization algorithms. The most significant benefit of these algorithms over the other optimization algorithms is that they do not postulate intricate operations and mathematical relationships such as integrals and derivatives [12-13]. These algorithms are either modeled on the foundation of the biological processes and exchanges of organisms (such as ants, particles, genetics, etc.) or human socio-political exchanges and treatments (such as colonial competition algorithms or teacher-learning based optimization) [14-15]. The PSO algorithm is also modeled based on the search for appropriate lodging by particles. This algorithm was suggested and developed in 1995 by a common study of Eberhart and Kennedy based on the motion of fish and particles on the basis of the two axioms of artificial life and evolution. In

similar other evolutionary algorithms, algorithm begins with a collection of particles of a matrix with a completely random position. Any particlein this matrix is called a particle, and these particles can jump in the nth - perspective space (n is the number of variables in the optimization problem). And at each step, their new situation is updated based onthe previous personal experiences and the situation of their proximities. The strength of each particle of this set of particles is defined by the following vector [16-18]:

$$\mathbf{Xi= [X_{i1}, X_{i2},\ldots\ldots X_{in}]^T} \in \ S \quad (1)$$

In this regard, S is the search space and Xi is the position of each particle intheiteration i algorithm. Each particle has a velocity at any step. Therefore, the velocity vector of all particles is defined by relationship 2 [16-18]:

$$\mathbf{Vi= [V_{i1}, V_{i2},\ldots\ldots V_{in}]^T} \in \ S \quad (2)$$

The best personal position that each particle has from the beginning toi step is called the best personal position and is defined for all particles by the following vector in each step [16-18]:

$$Pi= [Pi1, Pi2,\ldots\ldots Pin]T \ \in \ S \quad (3)$$

Based on the relationships and definitions described above, the rate and speed of each particle at each step of repetition is calculated and updated by the following relationship [16-18]:

$$\vec{v}_i^{k+1} = w\vec{v}_i^{k} + c_1 r_1 \times (\vec{p}_i - \vec{x}_i^{k}) \ + c_2 r_2 \times (\vec{p}_g - \vec{x}_i^{k}) \quad (4)$$

$$\vec{X}_i^{k+1} = \vec{V}_i^{k+1} + \vec{X}_i^{k} \quad (5)$$

In this regard, the updated speed of the particle is in the iteration of k + 1 and the previous velocity and location of the particle respectively. It is also the best I-thparticlelocation ever as well as the location of particle that has the p-best between particles. Here c1 and c2 are fixed coefficients and are usually 2. If the quantity of c1 increases, the particle tends to follow the search around its best personal location. However, if c2 is higher than c1, the tendency of the particle is to probearound the global location. Hence, it is better to assimilate the procedure of choice between these two parameters. The coefficient w is known as the inertial weight coefficient. This coefficient specifies the impact of the previous velocity on the new velocity. If the low w coefficient is c, the search step is short and consequently, the search space is small and of course, the search accuracy is increased. However, if theselected number is large, the search step and the search space for each particle will be longer but the search accuracy will be lower. r1 and r2 are two random numbers between zero and one that gives a random nature to the search pattern. In many cases, the w coefficient is fixed and about about0.9. However, in some cases it is linear and a function of program repetition. Sofirst,a large search is selected to enlarge the search space at the beginning of the search. Then, with increasing Iteration pattern, its value decreases so that the further we go, the more accurate the search accuracy. Although this method gives a more accurate answer than the choice of w with a constant value, it still cannot be applicable in all engineering issues. Therefore, it is then selected by fuzzy rules in a comparative manner. If the target function is close to the optimal value, the coefficient w is small and if it is far away, the coefficient w is selected. In addition to the coefficient w, the coefficients c1 and c2 will be selected by comparative relationships according to what will be mentioned in the next section. As mentioned above in the particle cluster algorithm, particles are inclined to follow a search pattern thatcan obtain the best personal and global location at each stage. This causes premature convergence of the algorithm because the actual main optimal point may be far from these two points. To overcome this problem, in this article we use the mutation operator to prevent particles from getting entangled in the optimum local point.

d. Modification of the Classic PSO Algorithm with Mutation Operator:

As mentioned before, the particle aggregation algorithm, despite the genetic algorithm, does not have a mutation operator and always tries at every step to find the search around the two points of the best personal and overall position to continue the same step. This phenomenon can lead to two demerits. First, the algorithm may experience premature convergence. This means that it is entangled in an optimal local location, in other words, the population loses its diversity. Second, the response varies from program to program since the final response depends almost on the randomly selected primary population. Hence, in this article, we use the mutation operator to overcome these two problems. Sincethe mutation is a powerful tool in improving particle population diversity. In this article, we will use a new mutation operator. First, five vectors are randomly selected from the previous population in each repetition of the program (H4, H5 H1, H2, H3,) so that H4 ≠ H5 ≠ H1 ≠ H2 ≠ H3. Now the jump operator selects the new position of the particle as follows:

$$Xmut = XH1 + \beta1\ (XH3- XH2) + \beta2\ (XH5- XH4) \quad (6)$$

Here, the coefficients β1 and β2 are supposed as mutation coefficients, the value of which should be selected experimentally in the range0 <β <1. In the following, the value of the position of each particle of the following relation is calculated:

$$X_{new,i} = \begin{cases} X_{mut,i}, if(rand < crossover) \\ X_i \text{Otherwise} \end{cases} i = 1,2,\ldots\ldots n \quad (7)$$

In this article, the crossover value is calculated as 0.2

e. Determining the Coefficients of c1 and c2 in a Comparative Manner:

As mentioned before, the coefficients C1 and C2 in the classical PSO algorithm are considered constant and equal to the value of 2. In some papers, these coefficients change linearly over different iterations. Increasing or decreasing these coefficients, in addition to directing the search around a particular point, can reduce or increase search space. As the matter of increasing or decreasing the weighting coefficient of inertia w. Experimental results show that such a choice for these coefficients prevents accurate response. For this reason, in this article, these coefficients are determined comparatively using the following relationship.

$$c_1, c_2 = 1 + [1 + exp(-\frac{G\_Best\_Valve}{G_0})^n]^{-1} \quad (8)$$

In this case, n = 2 and G0 are equivalent to Gbest in the first repetition. Notice that the smaller the Gbest in the current repetition, the closer we get to the answer. So its value is reduced to increase the accuracy of the search. However, if Gbest is a large number, the answer is far from the optimal global answer and makes the search space bigger.

f. Fuzzy Rules for Diagnosing the Inertial Coefficient W:

The weight factor W has a huge impact on the velocity of each particle at the current stage, so increasing this factor increases the velocity. Since it is supposed that in relationship number 5, the amount of each displacement is considered one second, so the higher the velocity, the higher the particle displacement in one step, and consequently, the search space is large and its accuracy is decreased. The opposite is true. Hence, an appropriate balance must be taken into account in selecting this particle. In this article, this equilibrium is performed using fuzzy rules and ifs. The best choice is to match the w coefficient to whether Gbest is close to or far from each step of the desired Gbest using fuzzy logic. Here, the values of w and NFV, which are defined below, are the inputs of the fuzzy inference motor and its output is Δw [19-20].

$$NFV = \frac{(FV - FV_{min})}{(FV_{max} - FV_{min})} \quad (9)$$

Here FV is the Gbest level in the current step and FVmin is the Gbest level in the first repetition and FVmax is a very large number. Usually, the W coefficient must be between 0.9 and 0.4. Since the correction of the W factor during the implementation of the program may be increasing or decreasing, both positive and negative corrections are essential for this coefficient. In this research, a small number with a value of 0.1 is regarded, which is added and subtracted by the W factor.

$$\omega^{k+1} = \omega^k + \Delta\omega \quad (10)$$

Here Δw is asimilar correction value and is equal to ± 1. Of course, sometimes the value is zero and its status is suggested according to Table 1. Notice that Gbest values must be expressed as membership functions to attain an optimal value for the weight factor W. In this article, it is recommended that triangular membership functions be selected so that they have three states:
Large or L, small or M, and medium or M. Also, the fuzzy model outputs, as shown in Table 1, have three values of PE ((+0.1, NE ((-0.1) or, ZE (0). As shown in Table 1, the 9 states may be based on different values of NFV and W occur. If both NFV and W are small, there is no need to change w because on the one hand,Gbest has reached the optimal level and on the other,hand it is not possible to decrease W so much that it excels the permutable limit. If the NFV is low and the W is medium, you can still reduce the W by 0.1 to increase the search accuracy. If the NFV is low and the W is high, you can reduce the w by 0.1 as much as before. Here the relationship between inputs and outputs is shown in Table 1. Also, the triangular membership functions are represented in Figure 4. These functions are used to get the input and output variables.

Table 1: Fuzzy rules of the input and output variables

ΔW		W		
		S	M	L
NFV	S	ZE	NE	NE
	M	PE	ZE	NE
	L	PE	ZE	NE

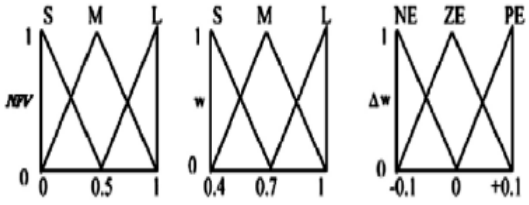

Fig. 4. The membership functions.

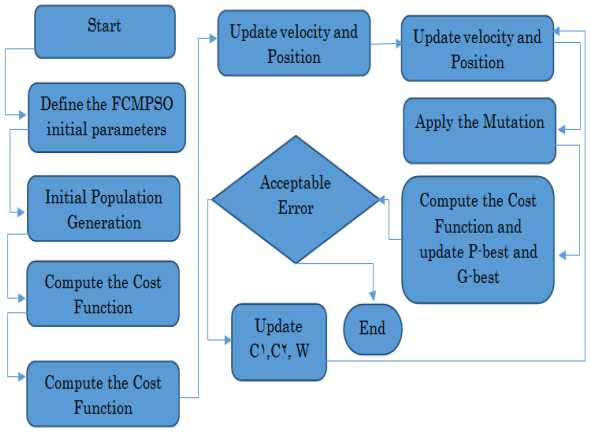

Fig. 5. Flowchart of FCMPSO algorithm

g. Experimental Data

KDDCUP99 [24] and NSL-KDD are the most commonly used datasets in intrusion detection research. We used theNSL-KDD intrusion dataset which is available in CSV format for model validation and evaluations. The dataset composes of the attacks shown in Tables2 and 3 and identified as a key attack in IOT computing. Sherasiya and Upadhyay [25] pointed out that IOT objects are also exposed to such types of attacks, and the data that IoT objects exchange are of the same value and importance, or occasionally more important than a non-IoT counterpart.

h. The Objective Function:

In this research, to model the attack detection system and anomalies, we used the multilayer perceptron neural network structure as ML. additionally we trained the neural network using BP algorithms, classical particle algorithms, modified particle algorithms with FPSO (fuzzy PSO), FCPSO (Fuzzy comparative PSO) and FCMPSO (Fuzzy combinations. Moreover, we used the sigmoid function as the last layer of the neural network according to the following formula.

$$a(z) = \frac{1}{1+\exp(-z)} \quad (11)$$

The accuracy of the suggested model is calculated based onthe correct detection of the model attained by the neural network and by the following relationship:

$$Accuracy = \frac{\text{True Positive+ True Negative}}{\text{True Positive+ True Negative+False Positive + False Negative}} \quad (12)$$

Since the particle algorithm inherently minimizes the target function, the following function should be defined to increase the accuracy of the target function:

$$\text{Cost Function} = - \text{ accuracy} \quad (13)$$

3- Result and Discussion

As referredto in the previous section, the PSO algorithm is a powerful algorithm for finding optimal points in complex and multi-purpose problems. Hence, in this article, theneural network has one hidden layer with 15 neurons and training is done by the PSO algorithm. However, the classic model of this algorithm has a number of coefficients that if selected consistently decrease particle diversity and premature convergence, resulting in localized optimal locations. So, in this paper, these coefficients c1, c2 are comparatively diagnosed using exponential relationships. Additionally, the weighted coefficient of inertia is determined using rolls and fuzzy logic rules. Also, since this algorithm, unlike the genetic algorithm, did not have a mutation operator, it led to the search for the best personal position or the best global position at any stage, so we suggested adding a new mutation operator to the algorithm's function. This operator is expected to curb the algorithm from getting trapped in the optimal local locations. So, we used the combination of the above methods and taught them the neural network and compared the outputs. Figure 5 shows the accuracy level for different neural network training methods. Here we suppose that the maximum repetition is equal to 50 and also the number of particles is equal to 40. As represented in the figure, neural network training by classical PSO algorithm is much more optimal than training by BP algorithm. Moreover, as expected, the classic PSO algorithm was entangled at the local optimal point, and the combination of FPSO, FCPSO, and FCMPSO gave more accurate responses. Also, the combination of the mutation operator with the classic PSO algorithm gives good results. Figure 7 shows the convergence speed of different algorithms drowned on the iteration of the algorithm for diagnosing different attacks. As shown in the figure, the FCMPSO algorithm, in addition to being much more accurate, has a better convergence pace. So, this algorithm is a very optimal algorithm to increase the accuracy and speed of attack detection.

From theDos detection picture, we can see that when the number of trainings exceed 35 times, the Classic ANN curve is basically stable, and with the increasing of the number of trainings, the accuracy rate no longer increases significantly. In this method, the performance accuracy of the algorithm does not exceed74%. In contrast to this method is the ANN-FCMPSO algorithm. This method has higher accuracy (99%) and achieves faster response. As shown in this figure convergence point is 26 and the point of this sentence is that FCMPSO algorithm is faster than theprevious algorithm. Moreover figure 7 shows that by applying any corrective methodology in PSO algorithms such as ANN FPSO and ANN FCPSO, accuracy and convergence speed improved simultaneously. Another matter is that among the four attack type, these methods give the best performance to the Dos attack.

Lastly, in Figure 8, we show the accuracy of the PSO and FCMPSO algorithms after running the program 20 times to detect Dos attacks. As shown in the figure, the FCMPSO algorithm is more dependable than the PSO algorithm. Since in different performances, the program represents relatively the same answers.

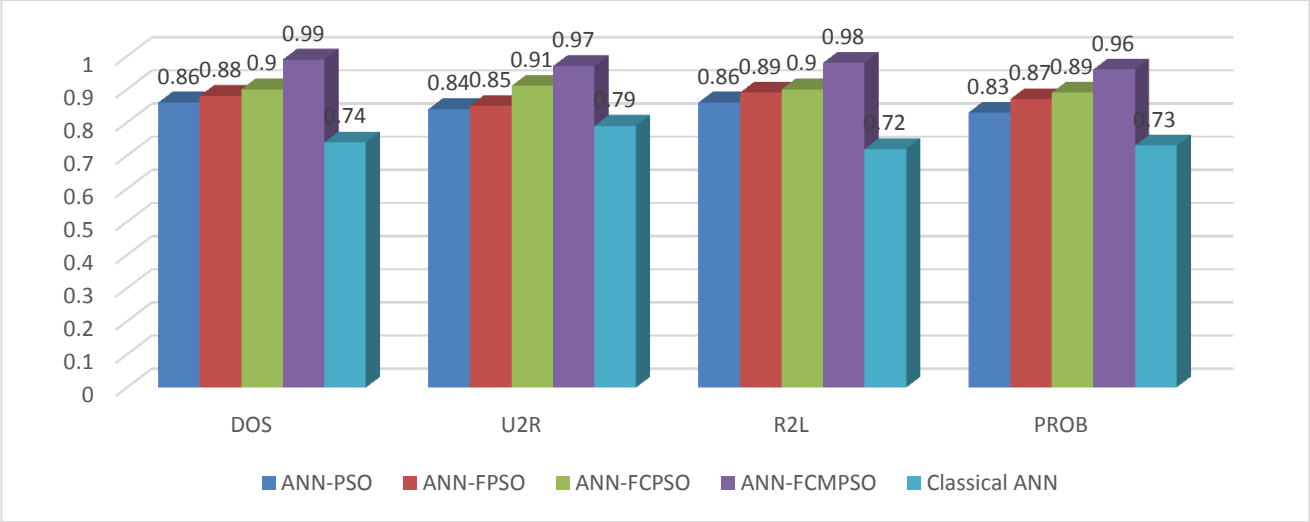

Fig. 6: accuracy for different machine learning algorithm

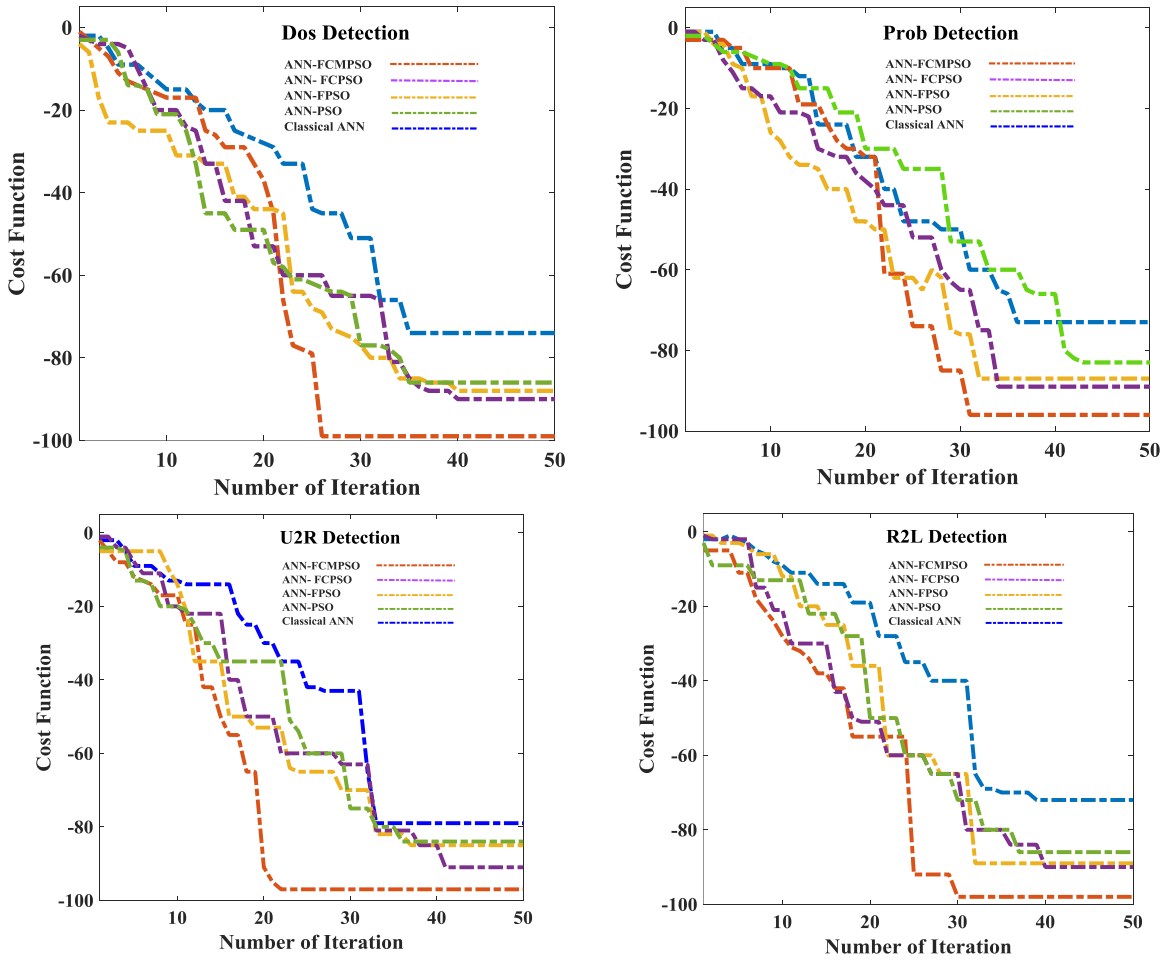

Fig. 7. convergence characteristic of proposed method in different attack detection

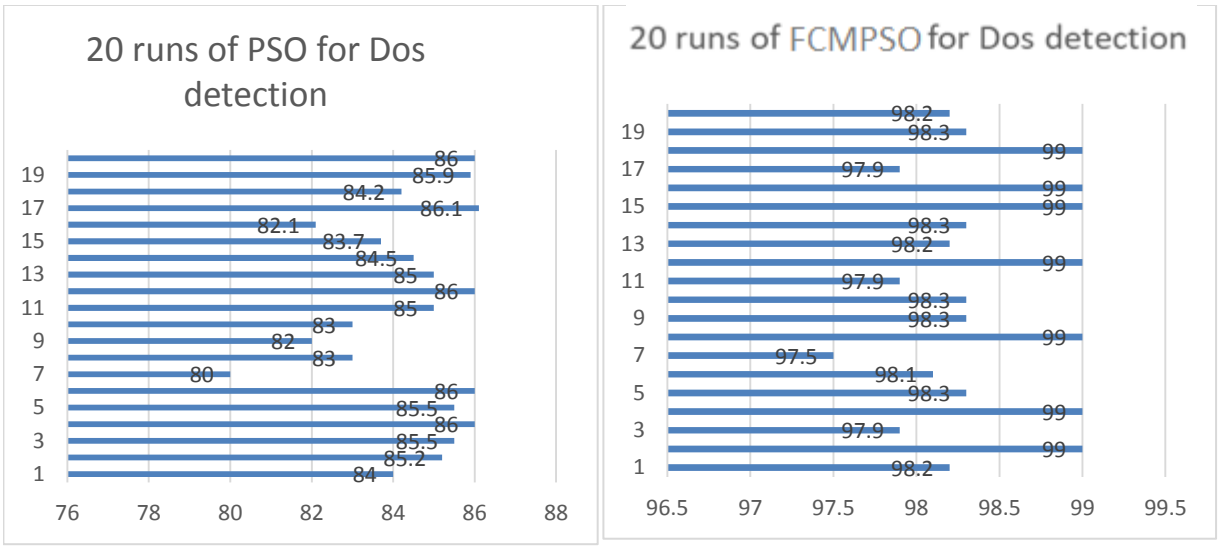

Fig. 8. Accuracy for 20 runs of the left algorithm: ANN-PSO right: ANN FCMPS

Table 2: Input parameters of Neural Network

S/N	Name	Type	S/N	Name	Type
1	duration	Continuous	25	serror_rate	Continuous
2	protocol_type	Symbolic	26	srv_serror_rate	Continuous
3	service	Symbolic	27	rerror_rate	Continuous
4	flag	Symbolic	28	srv_rerror_rate	Continuous
5	src_bytes	Continuous	29	same_srv_rate	Continuous
6	dst_bytes	Continuous	30	diff_srv_rate	Continuous
7	land	Symbolic	31	srv_diff_host_rate	Continuous
8	wrong_fragment	Continuous	32	dst_host_count	Continuous
9	urgent	Continuous	33	dst_host_srv_count	Continuous
10	hot	Continuous	34	dst_host_same_srv_rate	Continuous
11	num_failed_logins	Continuous	35	dst_host_diff_srv_rate	Continuous
12	logged_in	Symbolic	36	dst_host_same_src_port_rate	Continuous
13	num_compromised	Continuous	37	dst_host_srv_diff_host_rate	Continuous
14	root_shell	Continuous	38	dst_host_serror_rate	Continuous
15	su_attempted	Continuous	39	dst_host_srv_serror_rate	Continuous
16	num_root	Continuous	40	dst_host_rerror_rate	Continuous
17	num_file_creations	Continuous	41	dst_host_srv_rerror_rate	Continuous
18	num_shells	Continuous			
19	num_access_files	Continuous			
20	num_outbound_cmds	Continuous			

Table 3: Output Parameters of Neural Network (Attack Type)

S/N	Name	Type
1.	Back	dos
2.	buffer_overflow	u2r
3.	ftp_write	r2l
4.	guess_passwd	r2l
5.	imap	r2l
6.	ipsweep	probe
7.	land	dos
8.	loadmodule	u2r
9.	multihop	r2l
10.	neptune	dos
11.	nmap	probe
12.	perl	u2r
13.	phf	r2l
14.	pod	dos
15.	portsweep	probe
16.	rootkit	u2r
17.	satan	probe
18.	smurf	dos
19.	spy	r2l
20.	teardrop	dos
21.	warezclient	r2l
22.	warezmaster	r2l

4-Conclusion

In this study, we used modified PSO and PSO algorithms to train the neural networkto model the IOT network attack detection. We showed that meta-heuristic algorithms can be a more effective method than classical education systems. In addition, we have shown that the PSO algorithm has coefficients that, if not properly adjusted, lose their efficiency and cannot be suitable for neural network training methods. The correction model proposed in this paper is the simultaneous combination of a PSO algorithm with a fuzzy system and a mutational and adaptive operator. The suggested ANN-FCMPSO algorithm is about 97% (99% for Dos type attack, 97% for U2R, 98% for R2L and 96% for PROB), and the accuracy for the PSO-ANN algorithm is about 86%.

References

1- Haji, Saad Hikmat, and Siddeeq Y. Ameen. "Attack and anomaly detection in IOT networks using machine learning techniques: A review." Asian Journal of Research in Computer Science (2021): 30-46.
2- Aversano, Lerina, et al. "Effective Anomaly Detection Using Deep Learning in IoT Systems." Wireless Communications and Mobile Computing 2021 (2021). (+)
3- Khan, Arshiya, and Chase Cotton. "Detecting Attacks on IoT Devices using Featureless 1D-CNN." 2021 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2021.
4- Bello, Ibrahim, et al. "Detecting ransomware attacks using intelligent algorithms: recent development and next direction from deep learning and big data perspectives." Journal of Ambient Intelligence and Humanized Computing 12.9 (2021): 8699-8717.
5- Foley, John, Naghmeh Moradpoor, and Henry Ochen. "Employing a Machine Learning Approach to Detect Combined Internet of Things Attacks against Two Objective Functions Using a Novel Dataset." Security and Communication Networks 2020 (2020).
6- Ullah, Imtiaz, and Qusay H. Mahmoud. "Design and development of a deep learning-based model for anomaly detection in IoT networks." IEEE Access 9 (2021): 103906-103926.
7- Syed, Naeem Firdous, et al. "Denial of service attack detection through machine learning for the IOT." Journal of Information and Telecommunication (2020): 1-22.
8- Manimurugan, S., et al. "Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network." IEEE Access 8 (2020): 77396-77404.
9- Churcher, Andrew, et al. "An experimental analysis of attack classification using machine learning in iot networks." Sensors 21.2 (2021): 446. (+)
10- Latif, Shahid, et al. "A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network." IEEE Access 8 (2020): 89337-89350.
11- Alkronz, EyadSameh, et al. "Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network." (2019).
12- Wang, Weilin, et al. "Estimation of PM2. 5 concentrations in China using a spatial back propagation neural network." Scientific reports 9.1 (2019): 1-10.
13- Mohammadi, Farzaneh, et al. "Modelling and optimizing pyrene removal from the soil by phytoremediation using response surface methodology, artificial neural networks, and genetic algorithm." Chemosphere 237 (2019): 124486.
14- Azimi, Yousef, Seyed Hasan Khoshrou, and MortezaOsanloo. "Prediction of blast induced ground vibration (BIGV) of quarry mining using hybrid genetic algorithm optimized artificial neural network." Measurement 147 (2019): 106874.
15- Cai, Jianghui, et al. "A Novel Clustering Algorithm Based on DPC and PSO." IEEE Access 8 (2020): 88200-88214.
16- Singh, Shakti, Prachi Chauhan, and NirbhowJap Singh. "Capacity optimization of grid connected solar/fuel cell energy system using hybrid ABC-PSO algorithm." International Journal of Hydrogen Energy (2020).
17- Devarasiddappa, D., M. Chandrasekaran, and R. Arunachalam. "Experimental investigation and parametric optimization for minimizing surface roughness during WEDM of Ti6Al4V alloy using modified TLBO algorithm." Journal of the Brazilian Society of Mechanical Sciences and Engineering 42.3 (2020): 1-18.
18- Qiao, Weibiao, Hossein Moayedi, and Loke KokFoong. "Nature-inspired hybrid techniques of IWO, DA, ES, GA, and ICA, validated through a k-fold validation process predicting monthly natural gas consumption." Energy and Buildings (2020): 110023.
19- Prithi, S., and S. Sumathi. "LD2FA-PSO: A novel Learning Dynamic Deterministic Finite Automata with PSO algorithm for secured energy efficient routing in Wireless Sensor Network." Ad Hoc Networks 97 (2020): 102024.
20- Kacimi, MohandAkli, et al. "New mixed-coding PSO algorithm for a self-adaptive and automatic learning of Mamdani fuzzy rules." Engineering Applications of Artificial Intelligence 89 (2020): 103417.
21- Jallal, Mohammed Ali, Samira Chabaa, and AbdelouhabZeroual. "A novel deep neural network based on randomly occurring distributed delayed PSO algorithm for monitoring the energy produced by four dual-axis solar trackers." Renewable Energy 149 (2020): 1182-1196.
22- Niknam, Taher, Ehsan Azadfarsani, and Masoud Jabbari. "A new hybrid evolutionary algorithm based on new fuzzy adaptive PSO and NM algorithms for distribution feeder reconfiguration." Energy Conversion and Management 54.1 (2012): 7-16.
23- Niknam, Taher, Hassan DoagouMojarrad, and Majid Nayeripour. "A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch." Energy 35.4 (2010): 1764-1778.
24- M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, ‘‘A detailed analysis of the KDD CUP 99 data set,’’ Proc. 2nd IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA), Ottawa, ON, Canada, Jul. 2009, pp. 1–6.
25- A, Alghuried, ‘‘A model for anomalies detection in Internet of Things (IoT) using inverse weight clustering and decision tree,’’ M.S. thesis, School Comput., Dublin Inst. Technol., Dublin, Republic of Ireland, 2017.