IJICTR
International Journal of Information & Communication Technology Research

Volume 13- Number 1 – 2021 (32 -39)

# Detection of E-commerce Attacks and Anomalies using Adaptive Neuro-Fuzzy Inference System and Firefly Optimization Algorithm

**Fereidoon Rezaei**
Department of Information Technology Management
Central Tehran Branch, Islamic Azad University
Tehran, Iran
f.rezaei@kish.ir

**Mohammad Ali Afshar Kazemi***
Department of Industrial Management
Central Tehran Branch, Islamic Azad University
Tehran, Iran
M_afsharkazemi@iauec.ac.ir

**Mohammad Ali Keramati**
Department of Industrial Management
Central Tehran Branch, Islamic Azad University
Tehran, Iran
m-Keramati@iau-arak.ac.ir

Received: 10 January 2021 - Accepted: 15 March 2021

*Abstract*—**Detection of attacks and anomalies is one of the new challenges in promoting e-commerce technologies. Detecting anomalies of a network and the process of detecting destructive activities in e-commerce can be executed by analyzing the behavior of network traffic. Data mining systems/techniques are used extensively in intrusion detection systems (IDS) in order to detect anomalies. Reducing the size/dimensions of features plays an important role in intrusion detection since detecting anomalies, which are features of network traffic with high dimensions, is a time-consuming process. Choosing suitable and accurate features influences the speed of the proposed task/work analysis, resulting in an improved speed of detection. The present papers utilize a neural network for deep learning to detect e-commerce attacks and anomalies of e-commerce systems. Overfitting is a common event in multi-layer neural networks. In this paper, features are reduced by the firefly algorithm (FA) to avoid this effect. Simulation results illustrate that a neural network system performs with high accuracy using feature reduction. Ultimately, the neural network structure is optimized by using particle swarm optimization (PSO) to increase the accuracy of attack detection capability.**

***Keywords—Firefly Algorithm; Attack Detection; Neural Network; PSO Algorithm***

## I. INTRODUCTION

E-commerce was realized in 1991 for the first time to facilitate electronic businesses and trades [1]. This system allows individuals, companies, and organizations to send electronic documents securely over the internet. Generally, this is performed by sending commercial documents such as purchase orders or invoices electronically. Moreover, increasing trends towards android applications and the predominant contribution of commercial audiences in social networks make this capability face different challenges. The extended scope of e-commerce applications leads to increased energy consumption, complicated management systems, massive data, high bandwidth requirements for data transmission, and

* Corresponding Author

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

high-speed processing systems. One of the significant challenges is to preserve privacy and information security [2-4]. Definitely, the satisfaction of e-commerce users is tied to coping with these challenges. Threats and attacks that occurred in the e-commerce area can be classified as 1- DOS attacks, 2- R2L attacks, 3- U2R attacks, and 4- PROB attacks.

One way to deal with attacks and threats in the broad e-commerce network is to resort to ML models and procedures [6-7]. As a branch of artificial intelligence (AI) technology, ML rests on machines learning from their own experiences and predictions based on them. ML algorithms are trained using a training dataset to produce the models required. When new data are introduced to the ML algorithm, the system can perform the prediction process based on produced model. One of the most popular, widely-used, and robust ML models and algorithms is the artificial neural network (ANN) [8-9]. The present paper aims to detect threats and attacks by employing a modified ANN. Neural networks are well-organized network structures modeled base on the function of the biological nervous system in the human body. Neural networks are composed of three input, intermediate, and output layers connected by neurons. Input neurons receive data in the neural network.

The intermediate layers and neurons, which may be multivalued, receive these data and then process and analyze them. This data transfer is continued as long as the input data reach the output layer. Also, a mathematical or computational model is used in the neural network for data processing based on the coupling approach of calculation. One of the classic types of ANN is the perceptron network. A multi-layer perceptron (MLP) network (deep) is formed by sequencing several perceptrons. It means that there are multiple layers of neurons in such a network. There is also an input layer, an output layer, and several layers of neurons between input and output layers in this network. A layer that is located between input and output layers is called the “Hidden Layer.” Overfitting is a predominant challenge in neural network applications [10]. Today, one of the interesting methods to overcome this difficulty is "Feature Reduction." The firefly algorithm is applied to select useful features for coping with this adverse effect [11]. Next, to increase the reliability of the neural network, the number of layers and neurons in the neural network is optimized by particle swarm optimization (PSO) to obtain the minimum error in detecting attacks and anomalies.

## II. Related work

In [20], the Firefly Algorithm (FA) based on particle filter method and packaging has been used in order to choose the features. The resulted features are placed under the C4.5 category and Bayesian network with KDD CUP 99 data set.

In [21], a review of existing techniques and their applications is executed in connection with NIDS tools. They also prepared a complete list of attacks associated with network intrusion detection systems (HIDS) and host-based intrusion detection systems (HIDS). Moreover, they emphasized the need for extracting the main features that are playing a major role in detecting the anomalies. Detection methods and the criteria used to evaluate the NIDS performance were discussed.

In [22], the NISD-based neural network along with an increasing algorithm was proposed that owns less calculation. They also illustrated the correlation between features and attacks in the form of language. The experiments were conducted using KDD CUP 99 data set.

Similar work was conducted in [23]. Using parametric methods inspired by online Adaboost, uncontrolled detection of anomalies has been conducted in order to detect anomalies with unlabeled data. Despite its advantages, their deployment has been encountered problems in the network real setting/environment [24].

In [25], a hybrid method has been proposed in order to overcome the cluster-based works, which is a mix of K-Medoids clustering and Naive-Bayes categorization. In their work, the clustering method was applied to the data in order to form the group, and then, the clustering method was used in order to categorize the objectives and detect intrusion in the network.

In [26] and [27], data mining techniques have been utilized in order to detect unknown and unknown patterns and attacks.

In [28], a model based on an enhanced firefly algorithm (FA) has been utilized in order to choose the feature. This model uses firefly optimization to choose the features and it is a mix of approaches, based on filter and packaging, in order to increase the optimizer approach. In addition to the feature subset, the new categorization has been utilized in order to confirm the performance of chosen subset. In this work, KDD CUP99 has been utilized.

In [29], the features are chosen and efficiently categorized in order to reach a rate of optimized detection. Moreover, Map Reduce (which is a programming model) has been utilized in order to choose the optimal subset with minimum calculation complexity.
In [30], IDS was formed by using Rough Set Theory (RST) and SVM, where RST was utilized in order to choose crucial features.

## III. Proposed Design Review

Fig. 1. indicates an overview of the proposed design structure and flowchart. As can be seen, the required data should be collected in the first step. NSL-KDD dataset is used here. In the second step, the preprocessing operations, including identical data removal, data normalization, feature engineering, and vectorization, are performed over data. The FA is applied to feature extraction and reduction in this paper. Next, the selected data are classified into

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

training data (80%) and test data (20%). Finally, the ANN models the system concerning the training data.

## IV. NSL-KDD DATASET

Dataset is a set of data collected about a specific subject, and it is widely used in data mining. On the other hand, it is a valuable and efficient tool for testing and evaluating algorithms designed in a specific field. As an example, KDD CUP 99 dataset is collected for testing the intrusion detection algorithms.

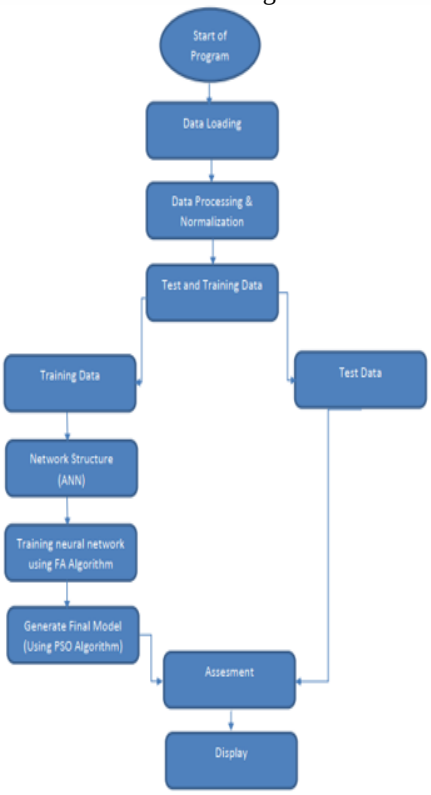


Figure 1. Flowchart of model derivation for detecting attacks from dataset.

This dataset has been prepared using the massive amount of data collected in the DIDE (Darpa Intrusion Detection Evaluation) project with the collaboration of the defense advanced research projects agency (DARPA) and Lincoln laboratory of MIT university. The dataset has been designed as a standard tool for evaluating intrusion detection systems (IDS) [12].
Hence, all records of this dataset have been labeled by cybersecurity experts so that it is convenient to recognize what specific class of attacks every record is belonged and also normal records. The database consists of two separate datasets: training dataset, which is also called learning set, and test dataset, which employs the learning set to detailed analysis of attack behavior and codification of effective and efficient rules. For the test and evaluation of the proposed algorithm, both datasets are applied. A well-known dataset adapted from KDD CUP 99 is NLS-KDD [12], prepared by detailed statistical analysis on KDD CUP 99 for coping with its intrinsic difficulties. Some of the advantages of NLS-KDD over KDD CUP 99 are as follows [12]:

1- There is no duplicate record in both the training and test datasets which increases the accuracy and efficiency of the data mining and machine learning algorithm and avoids the adverse effects of duplicate records on algorithm output.
2- In Both training and test datasets, the number of records is selected wisely and appropriately, increasing the speed of data mining and machine learning algorithms.

TABLE I. COMPARISON OF THE NUMBER OF RECORDS BETWEEN TWO DATASETS [12].

| Number of records in KDD CUP 99 | Number of records in NLS-KDD |
|---|---|
| 494021 | 125973 |

NLS-KDD includes 42 features or fields: 41 normal features related to network connections and 1 class feature with five different classes, including one normal class and four attack classes. Attack classes are R2L, U2R, DoS, and Prob [12].

## V. ARTIFICIAL NEURAL NETWORK

An artificial neural network (ANN) - usually referred to as a "Neural Network" - is a mathematical or computational model based on biological neural networks. In most cases, the neural network is an adaptive system, which changes its structure according to internal or external information flowed through the network in the learning step. Practically, neural networks are tools applied for nonlinear statistical modeling. They can be employed to find patterns governing data or model the complicated relationships between inputs and outputs. ANN, also named "Simulated Neural Network" or only "Neural Network," is an interconnected group of artificial neurons that utilizes a mathematical or computational model to process data. Its computational operations are conducted based on the connection approach of neurons. One of the classic ANNs is the perceptron network. The following Figure demonstrates both the biological and artificial neural networks [13-14].

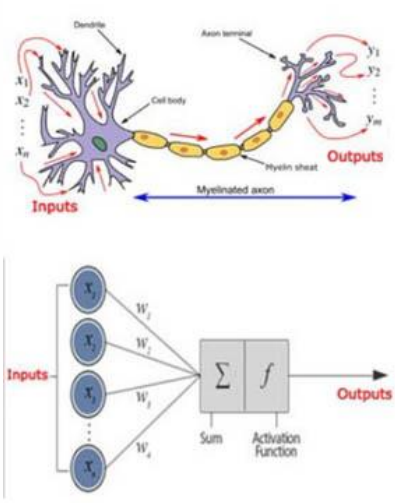


Figure 2. Neural network: top) biological, bottom) artificial.

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

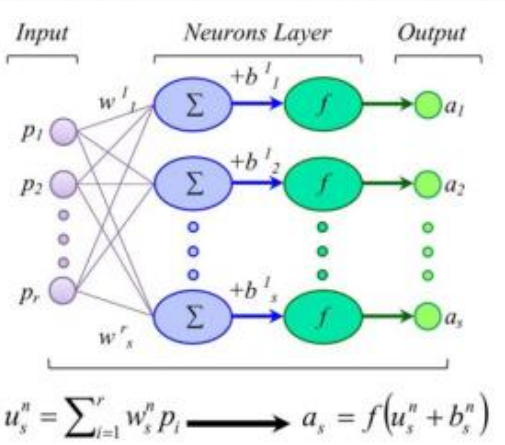


Figure 3. Perceptron neural network

Each node of neurons is trained by input dataset X: X1, X2, …, Xn. Inputs are multiplied by the weight coefficients W: W1, W2, …, Wn and summed with weight values b: b1, b2, …, bn. Then, the weighted values are given as input to the nonlinear activation function. Weight coefficients embedded in input connections are determined during the learning process. Each input value is multiplied by its associated weight coefficient and, finally, summed with each other. This value is given as input into the activation function. Typically, a sigmoid single-pole activation function is applied, and there are several training algorithms for a neural network that are different in terms of performance. The training algorithm is usually used because of its high speed, and it minimizes the training error performance. Only the weight values with the lowest training error are applied to deal with the error performance gradient at each iteration.

## VI. MULTILAYER PERCEPTRON NEURAL NETWORK (DEEP NEURAL NETWORK)

A multilayer perceptron neural network (deep) is formed by sequencing some perceptrons. The means that there are several layers of neurons in such a network. There is also an input layer, an output layer, and several layers of neurons between input and output in this network. A layer that is located between input and output layers is called the hidden layer. Layers near the input layer are called “Lower Layers,” and those near the output layer are “Upper Layers.” Except for the output, each layer has a bias. The network, including many hidden layers, is known as “Deep Neural Network.” In the 1990s, networks with more than two neurons were known as a deep neural networks. But nowadays, neural networks with more than hundreds of layers. Therefore, there is no clear definition for the term deep. Today, all neural networks are named deep, while some of them do not have this feature. The Figure below illustrates a multilayer neural network [13-14].

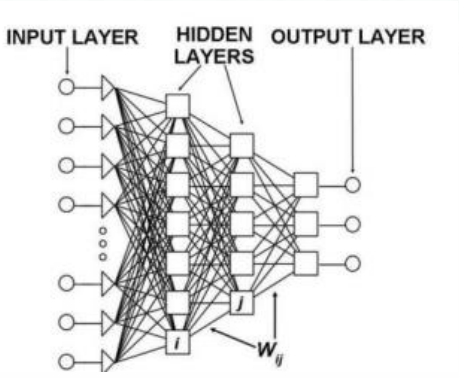


Figure 4. Multilayer perceptron neural network [14].

## VII. FIREFLY ALGORITHM

The firefly algorithm (FA) is inspired by the natural behavior of live creatures like other evolutionary algorithms. The algorithm proposed by Yang is modeled based on the firefly attraction process. This algorithm is simple and finds the optimum points without the requirements of complicated mathematical operators such as derivation and integral. This algorithm searches the optimum points based on the brightness criterion. It means that fireflies have random movements in the v-dimension space, where v represents the number of variables in the optimization problem. The attractiveness of each firefly is directly proportional to its objective function, and the less bright the firefly, the lower the attractiveness. Moreover, it should be noted that the distance between firefly and bait decreases its attractiveness. Therefore, the light intensity for firefly at a distance $\rho$ can be defined as follows [11]:

$$(1) \quad I = I_0 e^{-\gamma . r}$$

where $I_0$ is the light intensity at point r = 0 and $\gamma$ is the light attraction. Attractiveness of each firefly can be estimated based on the light intensity received by its neighbors. Therefore, attractiveness is defined as follows [11]:

$$(2) \quad \beta = \beta_0 e^{-\gamma . r^2}$$

where $\beta_0$ is the attractiveness at point r = 0. In this algorithm, each firefly moves toward fireflies with higher attractiveness. At each step, the displacement of attracted firefly i toward the more attractive (brighter) firefly j can be expressed as [11]:

$$(3) \quad X_i = X_j + \beta_0 e^{-\gamma . r^2}(X_i - X_j) + \alpha \in_i$$

where $\in$ is a random vector with Gaussian distribution. Also, $\alpha$ is the jump coefficient, which decreases linearly by increasing the number of iterations in this paper. It should be noted that since there is no firefly to attract the brightest firefly, it moves randomly. Moreover, it is assumed that fireflies have no sex, and each of them can attract other ones [11].

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

**Algorithm 1**. Pseudocode of the FA

1: Randomly generate *Np* fireflies (solutions) as an initial population $\{X_i | i = 1, 2, \ldots, Np\}$ ;
2: Calculate the fitness value $f$ of each firefly;
3: $FEs = Np$;
4: **while** $FEs < MaxFEs$ **do**
5: **for** $i = 1 : Np$ **do**
6: **for** $j = 1 : Np$ **do**
7: **if** $f(X_j) < f(X_i)$ **then**
8: Move firefly $X_i$ towards $X_j$ according to (3);
9: Calculate the fitness value of the new solution;
10: **end if**
11: **end for**
12: **end for**
13: $FEs = FEs + Np$;
14: **end while**

**Algorithm 1:** Pseudo-code of firefly algorithm [11]

## VIII. CLASSIC PARTICLE SWARM OPTIMIZATION

The PSO algorithm is one of the evolutionary optimization algorithms. The main advantage of this algorithm is the exclusion of complicated mathematical operations and relationships like derivative and integral. These algorithms are modeled inspiring from either the biological processes and interactions of live creatures (for example, ants, birds, genetic, etc.) or socio-political behaviors and interactions of people (for example, imperialist competitive or teacher forcing algorithm). PSO is also modeled based on the movement of birds in searching for the best habitat. Abarhart and Kenedi (1995) innovated this algorithm based on movements of birds and fishes considering two principles of artificial life and evolution. Like other evolutionary algorithms, PSO initially runs with a set of particles of a matrix at purely random positions. Every particle of this matrix is known as a bird. These birds can fly in an n-dimension space (n is the number of variables in the optimization problem), and their new positions are updated according to individual past experiences and their neighboring birds at each time step. The position of each particle from the group of birds is defined as the following vector [15-19]:

$$(4) \qquad X_i = [X_{i1}, X_{i2}, \ldots \ldots X_{in}]^T \in \ S$$

where S is the search space, and Xi is the position of each bird at iteration i. Each particle has a velocity at each step. Therefore, the velocity vector of all particles is expressed as follows [16-19]:

$$(5) \qquad V_i = [V_{i1}, V_{i2}, \ldots \ldots V_{in}]^T \in \ S$$

The most proper individual position of each bird from initial steps up to ith step is named "Personal Best Position," stated by the following vector at each time step [16-19].

$$(6) \qquad P_i = [P_{i1}, P_{i2}, \ldots \ldots P_{in}]^T \in \ S$$

Based on definitions and equations mentioned above, the velocity and displacement of each bird ate calculated and updated at each iteration by the following equations [16]:

$$(7) \qquad V_i^{\rho k+1} = WV_i^{\rho k} + c_1 r_1 \times \left(P_i^{\rho} - X_i^{\rho k}\right) + c_2 r_2 \times \left(P_g^{\rho} - X_i^{\rho k}\right)$$

$$(8) \qquad X_i^{\rho k+1} = V_i^{\rho k+1} + X_i^{\rho k}$$

where $v_i^{\rho k+1}$ is the particle velocity updated at iteration k+1, and $v_i^{\rho k}$ and $x_i^{\rho k}$ are the previous velocity and position of particle, respectively. Also, $p_i^{\rho}$ is the best position of ith particle so far and $p_g^{\rho}$ is the position of a particle that has the highest Pbest among all birds. c1 and c2 are the constant coefficients, which are usually assumed to be 2. If c1 increases, the particle tends to follow searching around its personal best position. On the other hand, if c2 exceeds c1, the particle tends toward searching around its global best position. Therefore, it is recommended to make a fair compromise between these constant values. Also, w is known as inertia weight. This coefficient represents the effect of the previous velocity on the new velocity. If it is selected as a low value, the search step will be short, which leads to a small search space and high accuracy of the searching process. Contrarily, its high value causes a long search step, extended search space, and low accuracy. r1 and r2 are two randomly selected values between 0-1, which provides the random nature of the searching process. In most studies, w is considered a constant value of 0.9 [15-19].

## IX. SIMULATION RESULTS

FA is a robust algorithm to find the optimum points in complicated and multi-objective problems, as mentioned in the previous section. For this purpose, the feature extraction is performed by this algorithm in the present study. It is worth noting that it is not required to evaluate output when FA is used for feature extraction. FA is applied to datasets for eliminating some features and select a class of features with a significant impact on the output to minimize the overfitting phenomenon. The results obtained by this algorithm are demonstrated in Table 1.

In this study, a deep neural network of 20 hidden layers is employed so that 80% of data provided for training and 20% as test data. The sigmoid function expressed in the following is used as the last layer of the neural network to evaluate the network performance and its training process with the BP algorithm.

$$(9) \qquad a(z) = \frac{1}{1 + exp^{(-z)}}$$

The accuracy of the proposed model is estimated according to true positives of the model obtained by the neural network as follows:

TABLE II. RESULTS OBTAINED BY APPLYING FA TO THE NSL-KDD DATASET

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NLS-KDD | 2 | 41 | 34 | 30 | 19 | 18 | 38 | 9 | 7 | 35 | 28 | 8 | 25 | 12 | 31 | 27 | 40 | 26 | 17 | 21 |

$$(10) \quad Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + Fale\ Negative}$$

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

The following Figures illustrate the results of applying the hybrid neural network to NLS-KDD datasets. After training the neural network, the test data results for 50 samples are as follows. In this diagram, the vertical axis values are expressed as follows: 1 for normal outputs, 2 for Prob attacks, 3 for U2R attacks, 4 for R2L attacks, and 5 for DoS attacks. As can be seen, by optimizing the neural network with PSO, the accuracy of detection has increased.


Figure 5. Neural network results without PSO.

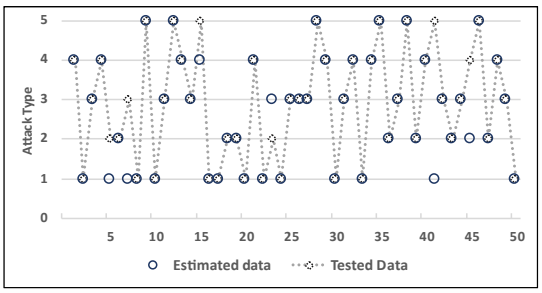


Figure 6. Predicting Neural network results with PSO

Fig. 5 and Fig. 6 show the results obtained by applying the neural network for attack detection without PSO and with PSO, respectively.

A comparison is performed between the proposed and fuzzy ARTMAP methods, whose results are illustrated in Fig. 7.

Tables (3) and (4), which is also named confusion matrix, are showing the exact statistics concerning the number of normal data and attacks before and after using the PSO algorithm. According to Table 5, TPR and FPR are calculated for two groups, namely normal data and attacks, and these show an increase in the accuracy after using the PSO algorithm.

TABLE III. CONFUSION MATRIX FOR ANN-FA

| | | **Actual** | |
|---|---|---|---|
| | | Normal | Attack |
| **Predicted** | Normal | 36037 | 7406 |
| | Attack | 3783 | 53552 |

TABLE IV. CONFUSION MATRIX FOR ANN-FA-PSO

| | | **Actual** | |
|---|---|---|---|
| | | Normal | Attack |
| **Predicted** | Normal | 37562 | 2894 |
| | Attack | 2258 | 58064 |

TABLE V. TRUE POSITIVE RATE AND FALSE POSITIVE RATE FOR ANN-FA-PSO AND ANN-FA

| **ANN-FA-PSO** | | **ANN-FA** | |
|---|---|---|---|
| TPR* | FPR** | TPR* | FPR** |
| 95.3% | 5.7% | 87.9% | 9.5% |
| *True Positive Rate | | **False Positive Rate | |

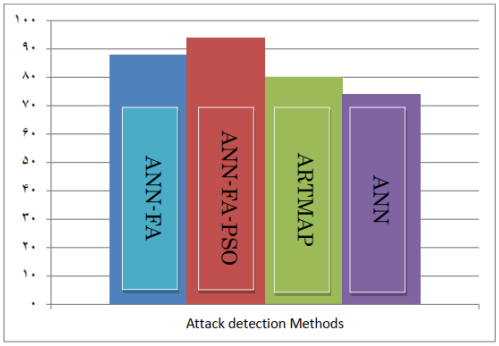


Figure 7. Comparison between results of different algorithms

As shown in Fig. 7, a combination of the neural network with firefly and PSO algorithms outperforms other ones.

## X. DISCUSSION

In [31], the PSO algorithm has been utilized for learning neural networks in order to detect attacks and anomalies in the Internet of Things system. Although the PSO algorithm has numerous advantages, it might, in some cases, reduce population diversity and lead to premature convergence. Hence, the TLBO algorithm has been utilized to solve this problem, increasing the accuracy of attack detection capability up to 90%. In our article, however, the firefly algorithm firstly reduced the features, leading to an increase in the accuracy of detection, and then the accuracy reached 94.3% after using the above-mentioned algorithm along with PSO. The proposed method is better than the one used in the above-mentioned article.

In [20], the Firefly Algorithm (FA) based on particle filter method and packaging has been used in order to choose the features. The resulted features are placed under the C4.5 category and Bayesian network with KDD CUP 99 data set. Here, the utilized method improved the accuracy of detection to an acceptable level. After the feature was reduced by the FA algorithm, a comparison between the final outputs illustrated that the PSO-based method can be a more suitable method in comparison to others.

Lacking the real data in order to detect the attacks in e-commerce is one of the limitations that this study faced. To diminish the effect of this problem, we used KDD CUP data were utilized since they are highly similar to e-commerce data in terms of features.

## XI. CONCLUSION

The present paper showed that the neural network has a relatively good performance in detecting e-

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

commerce related anomalies and attacks in terms of intrusion detection accuracy. Next, the firefly algorithm was employed to select a set of features to accelerate finding optimum solutions and improve the convergence rate of space searching, aiming to detect attacks and anomalies. This method increased the accuracy of attack and anomaly detection by 14%. Finally, the particle swarm optimization was utilized for optimizing the intended neural network, which resulted in a 6% increase in the accuracy of the intrusion detection.

## REFERENCES


[1] Darabpour, Mehrab (1397). Principles and Foundations of International Trade Law, Book 5: Competition Law, Representation, e-Commerce, and Intellectual Creations. Tehran: Ganje of Danesh. ISBN 978-622-6187-10-7.

[2] Hasan, Mahmudul, et al. "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches." Internet of Things 7 (2019): 100059.

[3] Kotenko, Igor, et al. "Attack detection in IoT critical infrastructures: a machine learning and big data processing approach." 2019 27th Euromicro International Conference on Parallel, Distributed and NetworkBased Processing (PDP). IEEE, 2019.

[4] 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).

[5] Mathan, K., et al. "A novel Gini index decision tree data mining method with neural network classifiers for prediction of heart disease." *Design automation for embedded systems* 22.3 (2018): 225-242.

[6] Doshi, Rohan, Noah Apthorpe, and Nick Feamster. "Machine learning ddos detection for onsumer internet of things devices." 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018.

[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] 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.

[10] Huang, Chiou-Jye, et al. "A novel hybrid deep neural network model for short-term electricity price forecasting." *International Journal of Energy Research* 45.2 (2021): 2511-2532.

[11] Peng, Hu, et al. "Enhancing firefly algorithm with courtship learning." *Information Sciences* 543 (2021): 18-42.

[12] S. Revathi, Dr. A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 ISSue 12, December-2013

[13] Marius-constantin, Popescu, Valentina E. Balas, Liliana Perescu-popescu, Nikos Mastorakis, "Multilayer Perceptron and Neural Networks", WSEAS Transactions on Circuits and Systems, ISSN: 1109-2734, Issue 7, Volume 8, July 2009

[14] Rana, A., Singh Rawat, A., Bijalwan, A., & Bahuguna, H. (2018). "Application of Multi Layer (Perceptron) Artificial Neural Network in the Diagnosis System": A Systematic Review. 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE). 978-1-5386-2599-6/18/$31.00 ©2018 IEEE

[15] Wang, Min, et al. "A calibration framework for the microparameters of the DEM model using the improved PSO algorithm." *Advanced Powder Technology* 32.2 (2021): 358-369

[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] Prithi, S., and S. Sumathi. "LD2FAPSO: 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.

[18] Kacimi, Mohand Akli, et al. "New mixed-coding PSO algorithm for a selfadaptive and automatic learning of Mamdani fuzzy rules." Engineering Applications of Artificial Intelligence 89 (2020): 103417.

[19] Jallal, Mohammed Ali, Samira Chabaa, and Abdelouhab Zeroual. "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.

[20] Selvakumar B , Muneeswaran K , Firefly algorithm based Feature Selection for Network Intrusion Detection, Computers & Security (2018), doi:https://doi.org/10.1016/j.cose.2018.11.005

[21] Monowar H. Bhuyan,D.K.Bhattacharyya and J.K. Kalita "Network anomaly detection: Methods,systems and tools," IEEE Commun. Surv. Tutor. 2014, 16, (1), pp. 303–336.

[22] G.Gowrison,K.Ramar,K.Muneeswaran and K.Revathi, "Minimal complexity attack classification intrusion detection system," Appl. Soft Comput., 2013, 13, (2), pp. 921–927 .

[23] Weiming Hu, Jun Gao, Yanguo Wang, Ou Wu, and Stephen Maybank, "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," IEEE Trans. Cybern., 2014, 44, (1), pp. 66–82.

[24] Jungsuk Song, HirokiTakakurb, yasuo Okabe, and Koji system," Inf. Sci., 2013, 231, (10), pp. 4–14.

[25] Deepak Upadhyaya and Shubha Jain, "Hybrid Approach for Network Intrusion Detection System Using K-Medoid Clustering and Naïve Bayes Classification," IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 3, No 1, pp 231-236, May 2013.

[26] B.UdayBabu, C.G. Priya and Vishakh, "Survey on intrusion detection techniques using data-mining domain", IJERT, 2014. Vol. 3.

[27] Vaishali B Kosamkar and Sangita S Chaudhari, "Data Mining Algorithms for Intrusion Detection System: An Overview," International Conference in Recent Trends in Information Technology and Computer Science (ICRTITCS), 2012.

[28] R. Success, D, Karunanidy, A. Dumka, J. Loganathan, "RFA Reinforced Firefly Algorithm to Identify Optimal Feature Subsets for Network IDS", International Journal of Grid and High Performance Computing, 12(3):5, February 2020, DOI: 10.4018/IJGHPC.2020070105

[29] Natesan P., Rajalaxmi R.R., and Gowrison G., "Hadoop based parallel Binary Bat Algorithm for Network Intrusion Detection", Springer,Int J Parallel Prog,PP. 1-20,2016.

[30] Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh, "Using Rough Set And Support Vector Machine For Network Intrusion Detection," International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 1, April 2009.

[31] M. Nazarpour, N. Nezafati, S. Shokouhyar, "Detection of Attacks and Anomalies in The Internet of Things System Using Neural Networks Based on Training with PSO and TLBO Algorithms", Signal Processing and Renewable Energy, ISSN: 2588-7327, eISSN: 2588-7335, December 2020, (pp. 81-94).


[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]

**Fereidoon Rezaei** holds M.Sc. in IT Engineering, Information Security. He is a Ph.D. student in IT Management, Business Intelligence. He is interested in Information Security, Business Intelligence, and Startups.

**Mohammad-Ali Afshar Kazemi** has a Ph.D. degree in Industrial Management, Operations Research. He is an Assistant Professor and a faculty member at the Islamic Azad University, Central Tehran Branch. His interests include Fuzzy Logic and Artificial Intelligence.

**Mohammad-Ali Keramati** has Ph.D. in Industrial Management. He is an Assistant Professor and a faculty member at the Islamic Azad University, Central Tehran Branch. He is interested in Operations Research, Statistical Analysis, and Fuzzy Systems.

[ Downloaded from ijict.itrc.ac.ir on 2025-01-21 ]