Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1440538, 10 pages
https://doi.org/10.1155/2022/1440538

Research Article

Artificial Intelligence-Based Security Protocols to Resist Attacks in Internet of Things

Rashmita Khilar,[1] **K. Mariyappan**,[2] **Mary Subaja Christo**,[3] **J. Amutharaj**,[4] **T. Anitha**,[1] **T. Rajendran**,[5] **and Areda Batu**[6]

[1]*Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India*
[2]*Department of Computer Science and Engineering, CMR University, Bangalore, India*
[3]*Department of Computer Science, School of Computing, SRM Institute of Science and Technology, Kattankulathur, India*
[4]*Department of Information Science & Engineering, RajaRajeswari College of Engineering, Bangalore, India*
[5]*Makeit Technologies (Center for Industrial Research), Coimbatore, India*
[6]*Department of Chemical Engineering, College of Biological and Chemical Engineering, Addis Ababa Science and Technology University, Ethiopia*

Correspondence should be addressed to T. Rajendran; rajendranthavasimuthuphd@gmail.com

Received 23 December 2021; Accepted 21 February 2022; Published 5 April 2022

Academic Editor: Fei Hao

IoT (Internet of Things) usage in industrial and scientific domains is progressively increasing. Currently, IoTs are utilized in numerous applications in different domains, similar to communication technology, environmental monitoring, agriculture, medical services, and manufacturing purposes. But, the IoT systems are vulnerable against various intrusions and attacks in the perspective on the security view. It is essential to create an intrusion detection model to detect and secure the network from different attacks and anomalies that continually happen in the network. In this paper, the anomaly detection model for an IoT network using deep neural networks (DNN) with chicken swarm optimization (CSO) algorithm was proposed. Presently, the DNN has demonstrated its efficiency in different fields that are applicable to its usage. Deep learning is the type of algorithm based on machine learning which used many layers to gradually extricate more significant features of level from the raw inputs. The UNSW-NB15 dataset was utilized to evaluate the anomaly detection model. The proposed model obtained 94.85% accuracy and 96.53% detection rate which is better than other compared techniques like GA-NB, GSO, and PSO for validation. The DNN-CSO model has performed well in detecting most of the attacks, and it is appropriate for detecting anomalies in the IoT network.

1. Introduction

Recently, IoT has acquired the interest of academic groups and of the ICT (information and communication technology) industry. IoT systems take on a number of facets of our daily lives, including health care, home environments, and transportation. Threats to IoT protection can cause serious privacy problems and economic damage [1]. IoT's development comes along with the emergence of numerous challenges. Any of these problems also arise as exceptions to the network anomalies, i.e., abnormal network traffic flow. A flash crowd, network failure, or variations in the network traffic may produce anomalies in terms of results, while attacks like probing and flooding attacks can also cause anomalies when it comes to security, attacks like Remote to Local (R2L) and User to Root (U2R) attacks [2].

Anomalies may be related to performance or security-related. Anomaly identification is an extremely important task for network operators in both situations. In particular, network operators needs an efficient method for quickly identifying abnormal unknown trends in traffic data to recognize irregular flows of traffic or the reasons of further handling anomalies [3]. In the sense of the IoT, a general description of an anomaly is the observable effects of an unpredicted

change in the condition of a system beyond its global or local norm. This description contains a number of significant observations about the existence of IoT data:

(i) Most of the data collected by an IoT system could be taken as "normal" since it reflects the typical operating qualities for that particular system

(ii) The definition of a system's "normal" operation can change for a number of reasons over time

(iii) The data produced by an IoT deployment shows only the actual processes that control the monitoring system [4]

In Figure 1, IoT networks consist of less cost sensors that were placed in three types of formats over a wide region, (1) centralized networks consisting of several, (2) decentralized networks, and (3) block-chain technology-based distributed networks. The sensors in these IoT networks perform the important roles in assuring the total efficiency of the IoT network [5].

There are instances in real-world datasets that are different from every other instance and called as anomalies. The identification of anomaly was to identify certain standards whose activity was deemed as abnormally correlated to normal nodes. The data leakage, fraud detection, and intrusion detection system are separate causes of anomalies. Detection of anomalies is used in a number of IoT domain regions, as presented in Table 1 [6–8].

1.1. Intrusion Detection. IoT devices are linked to the Internet and remain susceptible to attacks related to security. Incidents such as Denial-of-Service (DoS) and distributed DoS (DDoS) attacks create significant damage to the network. The major problem in IoT applications is identification and protection from such attacks that are mentioned in Table 1.

1.2. Fraud Detection. IoT networks are still vulnerable during logins or online purchases which can result in credit card details, bank data, or various sensitive details' theft.

1.3. Data Leakage. Sensitive data from file servers, databases, and various sources of data could leak to any external agency that not only contributes to data loss but further generates a threat which could compromise confidential system data. Suitable mechanisms of encryption will avoid such leaks.

Anomalies may be identified based on the point-wise, collective, or contextual forms. Point-wise anomalies tend to identify points that essentially deviate from the remaining data points and are utilized when series evolutions are not linear. Typically, it was utilized for detecting fraud.

Typical patterns of the time series like repeated pattern or forms from several IoT devices were identified collective anomalies. Shipping delay in the supply chain is very normal but if there are multiple delays, then it may take investigation and also collective study. Contextual anomalies are observed by taking into account the preceding type of information or context, like day of the week. Contexts are always very unique to a particular domain [9].

In Figure 2, the first process is to understand the type of the dataset collected. The next process is to distinguish the type of anomaly (i.e., point, contextual, and collective anomalies) from a predefined collection. The last process was to understand the training data availability for developing the anomaly detection model [10]. The novel contributions of this paper are structured as follows:

(i) Presented the anomaly detection model for security attack detection by means of DNN with the CSO algorithms. In this work, the optimization algorithm is proposed for optimizing the performance of the CSO algorithm

(ii) Deep learning is the class of machine learning algorithms which gradually extracted high-level feature from raw inputs using many layers. The UNSW-NB15 dataset was utilized for assessment of the anomaly detection model. This introduction part discusses the anomaly detection process in IoT and the concept of the proposed model

The remaining sections will be as follows: Section 2 discusses the relevant works on IoT anomaly detection, Section 3 discusses the proposed methodology, Section 4 presents the performance analysis of the proposed model, and Section 5 represents the conclusion of the work.

2. Related Works

Bagaa et al. proposed a security system for IoT based on machine learning model. This system leverages both Network Function Virtualization (NFV) and Software-Defined Networking (SDN) enablers for reducing various threats. This security system copes automatically with the expanding aspects of security associated with IoT domain. The system used the distributed data mining system, supervised learning, and neural network for developing this intrusion detection model. The NSL-KDD dataset used for evaluation and one class SVM technique was used to detect the attacks and obtained better detection accuracy. Overall, the performance was good and the results obtained were appropriate for this intrusion detection model [1].

Lawal et al. used different classification techniques like k-NN, J-48, and Naïve Bayes for classifying different attacks in the IoT intrusion detection model. For training and testing, the UNSW-NB15 dataset was utilized. Performance analysis of J48, k-NN, and NB classifiers utilizing the WEKA application was experimented on this dataset. Outcomes from the analysis demonstrated that k-NN achieved better accuracy and low FP rate in detecting abnormal and normal traffics, where J48 performed better in classification than NB and k-NN based on the attack classes [2].

Hoang and Nguyen proposed an anomaly detection model for IoT network traffic using PCA method. The PCA method was used for reducing higher data dimension. A new distance formula was proposed and implemented to derive formulas from past works. Based on those derivations, a new technique for anomaly detection in network traffic

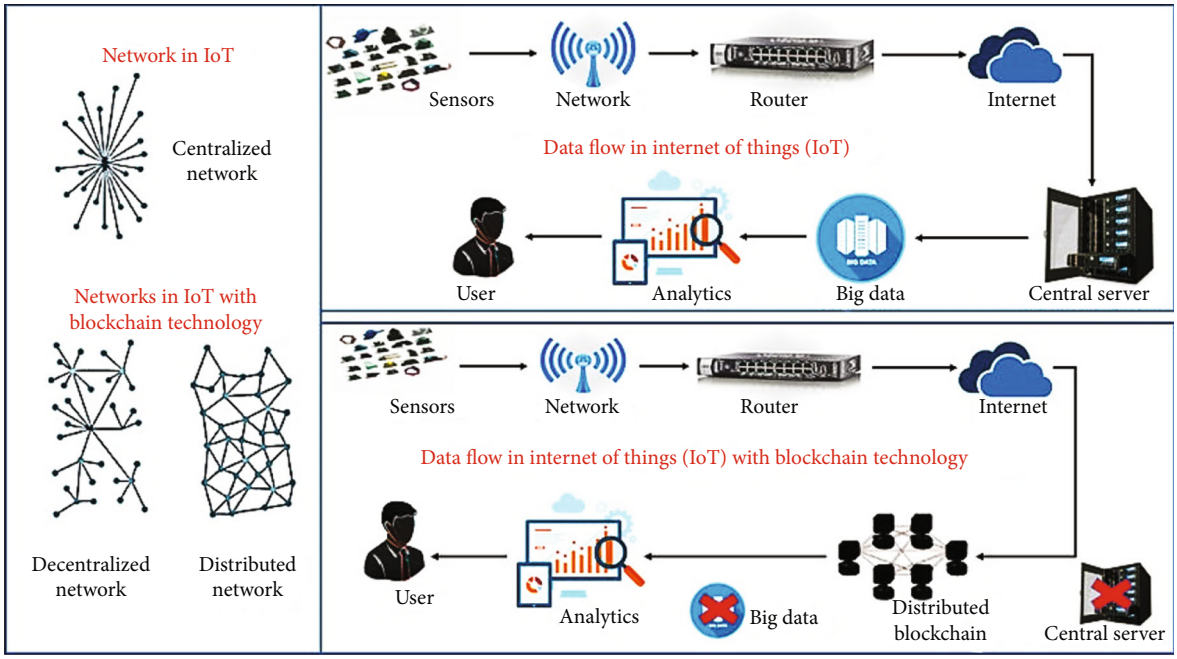

Figure 1: IoT network types [5].

Table 1: Anomaly detection in different IoT platforms.

Area	Anomaly detection	Advantages
IoT smart city	Health detection	Life saving
	Gas leakage	To save fuel
	Electricity leakage	To save energy
	Water leakages	Preventing water wastage
	Light bulbs broken	Maintenance times reduced
Network security	Intrusion detection DDoS attacks Fraud detections	For securing data
Industries	Surface inspection of devices	To solve defect on devices

was implemented and obtained appropriate results using new distance formula by reducing the computational overhead [3].

Sharmat et al. developed an anomaly detection model for IoT network using machine learning method. Artificial neural network and logistic regression techniques were used for classification. The Kaggle dataset was used for performance evaluation in this work. It was concluded that ANN was better than LR in case 1, and both have performed similar in case 2 [11].

Fahim and Sillitti proposed a hybrid learning anomaly detection using clustering and classification techniques. For clustering, Hierarchical Affinity Propagation (HAP) was used, and for classification, decision tree classifier CART technique was used. The model combines the data into anomaly and normal clusters by using HAP clustering. Then, the labeled data acquired from the clustering stage was used for training the CART and for classifying future unseen data. The model was able to automate the data labeling, which was an advantage to reduce human intervention [12].

Deep learning methods have been utilized by some researchers to detect network anomalies. The classification results and deep learning methods were compared in the study of [13], and the findings show that the deep learning technique performed better. However, they only looked at the categorization study on PortScan and regular network traffic. The actual network environment has many more network traffic kinds than two, making identification more challenging.

The signature-based techniques have a high detection accuracy and a fast detection speed; they are ineffective for detecting unknown network traffic. In comparison, anomaly-based methods are more adaptable and generalizable, and they

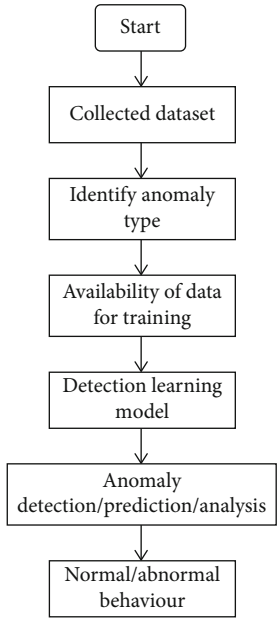

Figure 2: Anomaly detection flow chart.

perform well even when faced with classification tasks on unknown network traffics. Deep learning approaches, as compared to standard machine learning algorithms, have a quicker processing speed when dealing with large amounts of data and can learn the deep hidden representation of features with greater accuracy. So, in this research, a deep learning-based model with optimization algorithm is proposed.

3. Proposed Methodology

The anomaly detection model is proposed for an IoT network that uses a DNN with CSO. DNN has currently demonstrated its effectiveness in numerous fields that are important to its implementation. In Figure 3, deep learning is the algorithm which gradually extricates high-level features using multilayers from the raw input. For data collection, the UNSW-NB15 dataset was utilized for evaluating the proposed model. The integration of homogenous neural network classifiers results in a hybrid deep neural network-CSO model. The aggregation of classifiers is created by changing the activation of the neural network's weights and varying the input features.

3.1. Deep Neural Network. In this multilayer feed-forward DNN, the backpropagation technique is used. The backpropagation technique used supervised learning, while the approach was presented with input and output to be computed by the network and hence, the error is computed. The training started with random weight, and the purpose was to change them to minimize the errors. A neuron's weighted sum is calculated as

$$B_i(u, v) = \sum_{j-0}^{n} U_j V_{ij}, \tag{1}$$

where input sum U_j was multiplied by its relative weights, V_{ij}. The activation is just based on the weights and the inputs. If the identity will be the output function, hence, the neuron will be considered as linear. The used output function was sigmoid.

$$R_i(u, v) = \frac{1}{1 + f^{-B}(u, v)}. \tag{2}$$

The error is weight dependent and recommended for modifying to reduce the errors. The error functions for each neuron's outputs could be set to

$$F_i(u, v, d) = (R_i(u, v) - d_i)^2. \tag{3}$$

The result would be positive, and required targets would be bigger while the differences were bigger and smaller if the differences were smaller. The network errors would be simply a sum of all neuron errors in the output layer:

$$F(u, v, d) = \sum_i (R_i(u, v) - d_i)^2, \tag{4}$$

where R_i and d_i were the target output; the weight modified using the gradient descent method after finding this is the equation as follows:

$$\Delta v_{ij} = -\eta \frac{\partial F}{\partial v_{ij}}. \tag{5}$$

This equation can be interpreted as follows: the change of each weight was a constant negative eta (η); thus, learning rate wasη, multiplied by previous weight dependency on network error, which was a derivative of$F$in relation tov_{ij}.

The size of the correction would depend on η and the weight contribution to the function's error. That is, if the weight provides a great deal to the error, the correction is higher than it provides to the lower amounts. Equation (5) was utilized with a minimalized error before sufficient weights are established.

From now on, the F derivative was discovered in respect of v_{ij}. This is the objective of the backpropagation algorithm since it is important to achieve backward. Firstly, calculate the errors according to the outputs, with the derivative of$F$ from Equations (3) and (4) in relation toR_i.

$$\frac{\partial F}{\partial R_i} = 2(R_i - d_i). \tag{6}$$

According to activations, the output depended on weights from Equations (1) and (2), respectively. That could be noted from Equations (6) and (7):

$$\frac{\partial R_i}{\partial v_{ij}} = \frac{\partial R_i}{\partial B_i} \frac{\partial B_i}{\partial v_{ij}} = R_i(1 - R_i)u_j, \tag{7}$$

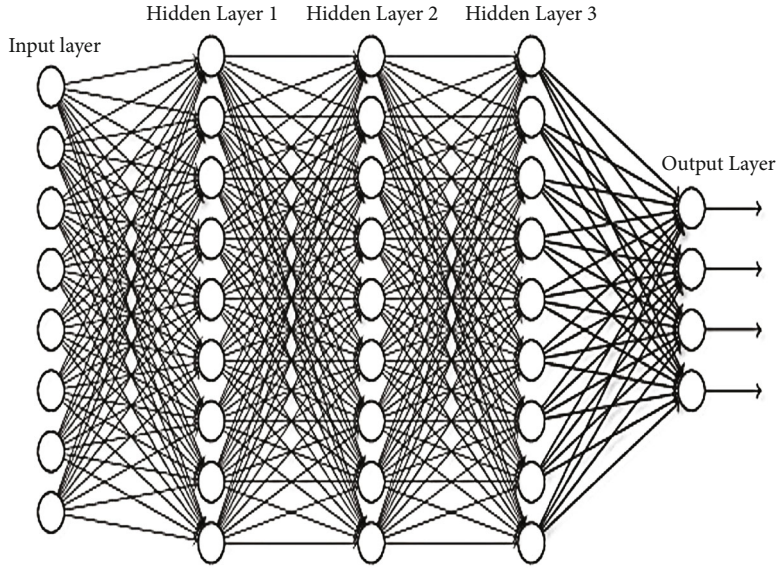

Figure 3: Architecture of deep neural network.

$$\frac{\partial F}{\partial v_{ij}} = \frac{\partial F}{\partial R_i}\frac{\partial R_i}{\partial v_{ij}} = 2(R_i - d_i)R_i(1 - R_i)u_j. \tag{8}$$

The adjustment will begin from Equations (5) and (8) for each weight:

$$\Delta v_{ij} = -2\eta(R_i - d_i)R_i(1 - R_i)u_j. \tag{9}$$

In Equation (9), in order to train the networks with an additional layer, some factors were required specifically on the training period that may be impacted with network architectures [13].

3.2. Chicken Swarm Optimization. CSO was an algorithm of bioinspired optimization. In the chicken swarm, it imitates the hierarchical orders and the chicken swarm behaviors. The chicken swarm could be categorized as several groups, containing a rooster and various chicks and hens. Various chickens followed various laws of movement. Under a particular hierarchical order, there are competitions between different chickens. Activities of chickens are by the values that follow the principles.

(1) Several groups are present in the chicken swarm. All groups have a predominant rooster, a few hens, and chickens

(2) How the chicken swarm can be divided into several classes and identification of chickens according to fitness value of chicken itself. The chicken with a higher fitness value will be carried out as rooster; each of that would be the group's head chicken. The chicken with low fitness value will be marked as chicks. The remainder is to be the hens. The hens choose randomly the party they want to live in. The mother-child link among the chickens and hens will be settled randomly

(3) The hierarchical structure, the close bond, and the bond between the mother and child within the group will remain constant. These conditions update many (G) timely steps

(4) Chicken tracks the rooster of their groups' mate to look for foods, although they may avoid eating their own food. Consider chickens poaching the best food found by others, accidentally. The chicks search around their mother (a hen) for food. A strong individual has an upper hand in a food competition

Chickens and chicken activities with the better fitness value may look for food across a wide range of distances. The chicken's movement ability is given in the following condition:

$$\Delta A_{i,j}^{c+1} = A_{i,j}^{c} * \left(1 + \text{Randn}\left(0, \sigma^2\right)\right), \tag{10}$$

$$\sigma^2 = \begin{cases} 1, & \text{if } f_i \le f_k, \\ \exp\dfrac{(f_k - f_i)}{|f_i| + \varepsilon}, & \text{otherwise, } k \in [1, N], k \ne i, \end{cases} \tag{11}$$

where $\text{Randn}(0, \sigma^2)$ was the Gaussian distribution with mean zero and standard deviations; σ^2 was utilized to prevent zero-division-errors. $K$is the index of rooster which was selected at random from the rooster groups, and f was the fitness values of related A. This phenomenon is formulated according to the following:

$$A_{i,j}^{c+1} = A_{i,j}^{c} + \text{S1} * \text{Rand} * \left(A_{i,j}^{c+1} - A_{i,j}^{c}\right) + \text{S2} * \text{Rand} * \left(A_{i,j}^{c+1} - A_{i,j}^{c}\right), \tag{12}$$

Initialize
repeat
Employ and order the fitness values of chicken using Equations ((10)) and ((11))
Isolate groups and select relations among chickens and hen using Equations ((12)), ((13)), and ((14))
Updating the chicken's solution till chicken's swarm find the better solutions using Equation ((15))
Memorizing the better solutions obtained so far
Until (Cycle = Max Cycle Numbers), Save best solutions
Assigning all networks input and outputs to DNN backpropagations
Initializing each weight from step 7
repeat
Presenting the patterns to the networks
Propagating the inputs forward through the networks
for all layers in the networks
for all nodes in the layers
Compute the weighted sum of the input to the nodes
Add threshold to the sum
Compute activation for the nodes
end
end
Propagating the error backwards through the networks
for all nodes in the output layers
Compute error signals
end
for all hidden layers
for all nodes in the layers
Compute node's signal errors
Updating every node's weighted in the networks
end
end
Compute Global Errors
Compute the Errors Function
end
While ((max numbers of iteration < than specified))

Algorithm 1

$$S1 = \frac{\exp\left(f_i - f_{r1}\right)}{(abs(f_i)} + \varepsilon, \tag{13}$$

$$\Delta S1 = \exp\left(\left(f_{r2} - f_i\right)\right). \tag{14}$$

The greater the difference between the fitness values of the two chicken, the lesser the S2 and the greater the distance between the positions of the two chickens. So the hens will not eat the food provided by other chickens quickly. The formula structure of S1 was different from S2 where there are competitions in a group. The chicks travel to search for food around their mother' it is expressed as

$$A_{i,j}^{c+1} = A_{i,j}^{c} + FL * \left(A_{m,j}^{c} = A_{i,j}^{c}\right), \tag{15}$$

where $A_{i,j}^{c+1}$ represents the location of the i^{th} chick's mom ($m \in [1, N]$). FL ($FL \in (0, 2)$) was the parameter, meaning the chicks will follow his mom to search foods. The differences were treated individually; the FL of every chick could select at random among zero and two [14].

The mathematical model of CSO could be comprehended in an accompanying manner: initially, verify the group structure, in particular the total of roosters, hens, chicks, and the mother hens; then, set determined identities for every chick; thirdly, set up the mathematical model by the identities of the chickens and their foraging laws; and finally, set a specific interval to update the relationship of chickens frequently. In the group, the number of roosters and chicks is smaller than that of hens, and their structures are generally simple. The number of hens is the largest, and the hens' structure is the most difficult in the group. In this way, the hen model will directly impact the performance of the CSO [15].

3.3. DNN-Based CSO Algorithm. Steps 3–5 develop the CSO algorithm and satisfy Equations (10)–(15) to enhance the weights in DNN.

4. Performance Analysis

Performance analysis and implementation of the proposed model are performed on a computer with Core i5 3.20 GHz CPU and 4 GB RAM in MATLAB 2017a. The proposed approach would be assessed using the output parameters such as accuracy, recall, precisions, F1-score, and detection rates [16–20]. The analysis of the performance of

Table 2: Selected features of the dataset.

Term	Types	Description
Srrcip	Nominals	IP addresses of sources
Dstipp	Nominals	IP addresses of destinations
Dsports	Integers	Ports numbers of destinations
Protol	Nominals	Protocols for transactions
Dura	Float	Overall duration records
Dbytess	Integers	Transactional byte of destinations to sources
iss_ftpp_login	Binary	If ftp sessions were accessed by users and passwords, hence 1 else 0.
cts_srsv_dsst	Integers	Total link that includes the similar services (14) and addresses of destinations (3) in 100 links depended on last times (26)
cts_dsst_ltsm	Integers	Total links of same addresses of destinations (3) in 100 links depended on the previous times (26)
cts_src_ ltsm	Integers	Total link of same addresses of sources (1) in 100 links depended on the previous times (26)
cts_ssrc_dsport_ltsm	Integers	Total link of same addresses of sources (1) and destination ports (4) in 100 links according to the past times (26)

Table 3: Dataset traffic distributions.

Traffic label	Description/characteristic	Training record	Testing record
Worm	Intruder replicates itself to spread to other computers	130	44
Shellcodes	A small part of the code utilized as the payload in the exploitation of software vulnerability	1133	378
Backdoors	A method in which a system security is bypassed stealthily to access a computer or its data.	1746	583
Analyses	It includes various attacks of port scan, spam, and html file penetrations	2000	677
Reconnaissance	Contains all strikes that can simulate attacks that collect data	10491	3496
DoS	A malicious effort to make a network or server resource unavailable to users, usually by temporarily suspending or interrupting the host's services connected to the Internet	12264	4089
Fuzzer	Attempting to cause a network or program suspended by feeding it the randomly generated data	18184	6062
Exploit	The attacker knows of a security issue within an OS or a part of software and leverages that knowledge by exploiting the vulnerability	33393	11132
Generics	A method works against each block cipher, without considering about the block-cipher structure	40000	18871
Normal	Natural transaction data	56000	37000
Total		175341	82322

the proposed DNN-CSO approach will be compared with the other techniques such as GA-NB, GSO, and PSO.

4.1. Description of Dataset. The IXIA PerfectStorm application creates the raw network packet of the UNSW-NB15 dataset in the Cyber Range Labs of Australian Centre for Cyber Security (ACCS) to create the integration of true modern general operation and synthetic modern attack behaviors. Tcp_dump application was utilized to collect raw traffics over 100 GB (i.e., Pcap file). This dataset included nine attack types like Backdoor, Analysis, Exploits, Fuzzer, Shellcodes, DoS, Generics, Worm, and Reconnaissance. Bro-IDS and Argus were utilized, and 12 approaches were generated for producing 49 attributes overall [21]. The dataset was accessible from https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cyb-ersecurity/ADFA-NB15-Datasets/.

For training and testing, the dataset is divided into 70% for training and 30% for testing. The training sets contain 175341 instances, and testing sets include 82332 instances from various attack types and normal. In this analysis, just 12 attributes were chosen for performing the analysis from 49 attributes. The attributes chosen were cts-srv-dsst, scrips, cts-dsst-ltsm, cts-ssrc-dsport-ltsm, cts-ssrc-ltsm, dur, cts-dsst-ssrc-ltsm, dssport, dsbytes, dsstip, protos, and iss-ftps-logins as seen in Table 2. The traffic distributions of the dataset are represented in Table 3.

4.2. Performance Metrics. The accuracy was simply a subset of the model's performances. It is one of the performance indicators used to assess classification approaches. The following expression was used to compute the accuracy:

$$\text{Accuracy} = \frac{\text{TPV} + \text{TNV}}{\text{TPV} + \text{TNV} + \text{FPV} + \text{FNV}}. \quad (16)$$

Precision was defined as the positive prediction rates. It was described as proportions of correctly predicted positive

Table 4: Comparison of each attack by classifiers.

Attack class	GA-NB	GSO	PSO	DNN-CSO
Worms	58.21	62.83	65.50	80.53
Shellcode	85.96	88.93	91.67	93.05
Backdoors	48.29	52.84	54.27	59.47
Analysis	41.40	39.12	55.08	68.14
Reconnaissance	56.90	85.36	88.71	90.01
DoS	66.10	83.94	85.51	89.82
Fuzzers	50.57	66.23	54.18	70.36
Exploits	45.85	50.63	48.20	69.15
Generic	89.53	90.08	93.46	96.52
Normal	70.32	82.27	85.86	90.79

observation to totally predicted positive values. The following expression is used to compute precision:

$$\text{Precision} = \frac{\text{TPV}}{\text{TPV} + \text{FPV}}. \tag{17}$$

The recall was also known as the sensitivity. It was the ratio of each observation in the actual classes to the correctly predicted positive values. The following equation was used to compute recall:

$$\text{Recall} = \frac{\text{TPV}}{(\text{TPV} + \text{FNV})}. \tag{18}$$

The detection rate was the measure of the numbers of intrusion incidents. It reflects the total number of appropriate positive class predictions produced as the percentage of all predictions made. The DR was calculated by using

$$\text{DR} = \frac{\text{TPV}}{\text{TPV} + \text{FNV}}. \tag{19}$$

F1-score was the harmonic mean estimation of precision and recall. This metric, which was connected to accuracy, was ideal for measuring the performance detection of unbalanced data.

$$\text{F1 Score} = \frac{2\text{TPV}}{2\text{TPV} + \text{FPV} + \text{FNV}}. \tag{20}$$

The attack detected performance was assessed using the proposed approach and correlated with the various existing approaches like Genetic Algorithm with Naïve Bayes (GA-NB), Glowworm Swarm Optimization (GSO), and Particle Swarm Optimization as seen in Table 4. Ten types of attacks comprising normal attack labels were utilized for these performances of attack identification. The proposed approaches detected every attacking labelled with higher detection rates. The least performance model is GA-NB, and GSO and PSO were close and equivalent in the performances shown in Figure 4.

According to these characteristics, the proposed approach's assessment was based on the identification of

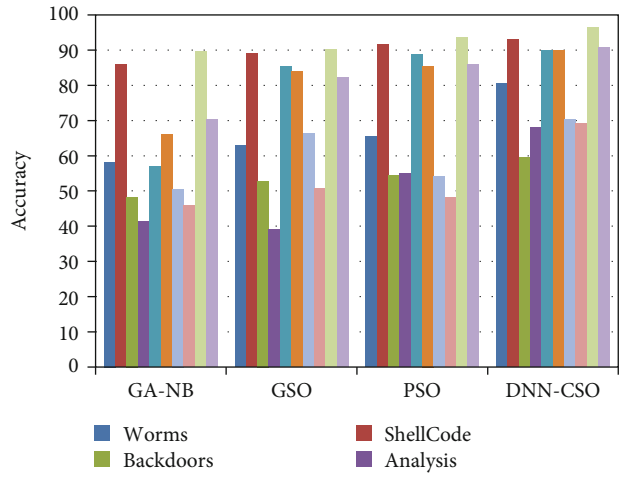

Figure 4: Accuracy comparison of attacks classified.

Table 5: Performance analysis comparison.

Method	Accuracy	Precision	Recall	F1-score
GA-NB	82.35	75.63	90.40	82.51
GSO	86.04	78.27	92.15	85.29
PSO	89.20	80.41	94.00	88.40
DNN-CSO	94.85	85.59	95.53	90.72

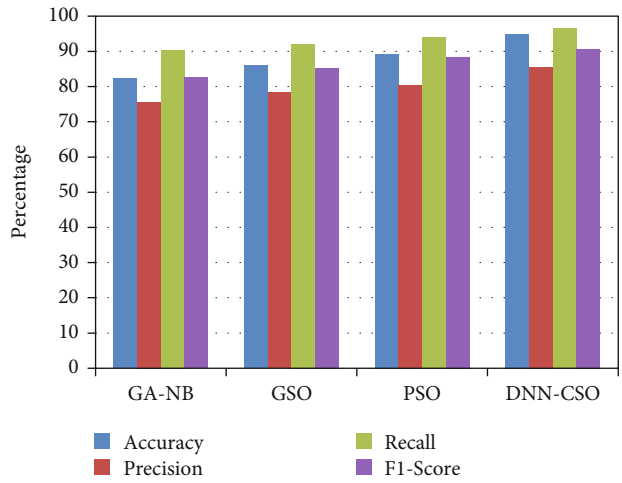

Figure 5: Comparison of performance analysis.

attacks in the input dataset. Accuracy was the appropriate detection range with each instance; detection rate was the detection ratio of classifier attacks; F1-score described the estimate of unbalanced samples; and recall reflected how many attacks the system returned. Precision referred to how many of the returning attacks were right. To validate the proposed DNN-CSO approach, the performance of several outcome parameters was assessed, as seen in Table 5.

The proposed method's performance was assessed by accuracy, detection rates or recall, precision, and F1-scores. As shown in Figure 5, the comparison of every performance

assessment of the approach was shown by demonstrating the difference among every classifier.

The DNN-CSO technique outperformed all other assessment criteria, comprising accuracy and detection rates. The DNN-CSO attained an accuracy of 94.85 percent, which was 5.6 percent to 12.5 percent greater than the other evaluated approaches. The proposed approach achieves a detection rate of 96.53 percent, which was 1.5 percent to 5.13 percent greater than other compared approaches.

5. Conclusion

Anomaly detection in IoT networks using deep neural networks with chicken swarm optimization algorithm was proposed. The DNN technique was used for feature selection and extraction of the dataset. The UNSW-NB15 dataset was used for generating the combinations of actual modern normal performances and synthetic modern attack behaviors in this model. Out of 49 features from the dataset, only 12 features were selected for the performance evaluation. Ten types of attacks comprising normal attack labels were utilized for these performances of attack identification. The proposed approach detected every attack label with higher detection rates compared with other techniques. The features of the dataset are effectively extracted by the DNN, and the CSO was used to classify and detect the attacks. For performance analysis, various parameters like accuracy, recall, precision, detection rates, and F1-score were evaluated. The DNN-CSO approach obtained the best performances in every evaluation term comprising detection rate and accuracy. DNN-CSO obtained 94.85% accuracy which was 5.6% to 12.5% improved than various compared approaches. The detection rates obtained by the presented approach was 96.53%, which was 1.5% to 5.13% greater than compared approaches. In the future, the proposed anomaly detection model can be used for detecting various attacks using different datasets for different network platforms like WSN, Cloud, and ad hoc networks.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

There is no conflict of interest.

Authors' Contributions

Authors, Rashmita Khilar, K. Mariyappan, Mary Subaja Christo and J Amutharaj are responsible for surveys and content writing and proofreading. Authors, Anitha T, Rajendran T and Areda Batu are responsible for algorithm design, development, and proofreading.

References

[1] B. Miloud, T. Tarik, B. B. Jorge, and S. Antonio, "A machine learning security framework for IoT systems," *IEEE Access*, vol. 8, pp. 114066–114077, 2020.

[2] A. L. Muhammad, A. S. Riaz, and R. H. Syed, "Security analysis of network anomalies mitigation schemes in IoT networks," *IEEE Access*, vol. 8, pp. 43355–43374, 2020.

[3] H. H. Dang and H. D. Nguyaen, "A PCA-based methods for IoT networks traffics anomaly detections," in *International Conferences on Advance Communications Technology*, pp. 381–386, Chuncheon, South Korea, 2018.

[4] C. Andrew, M. Goksel, and F. Zhong, "Anomaly detection for IoT time-series data: a survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2020.

[5] G. Anuroop, W. Tim, and A. Maia, "Anomaly detections model for detecting sensors fault and outlier in the IoT – a survey," in *International Conferences on Sensing Technology (ICST)*, pp. 1–6, Chuncheon, South Korea, 2019.

[6] T. Yu, Y. Sun, S. Nanda, V. Sekar, and S. Seshan, *RADAR: a robust behavioral anomaly detection for IoT devices in enterprise networks (CMU-CyLab-19-003)*, 2021.

[7] E. Apostol, C. Truică, F. Pop, and C. Esposito, "Change point enhanced anomaly detection for IoT time series data," *Water*, vol. 13, no. 12, p. 1633, 2021.

[8] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, 2021.

[9] G. Jyotheesh, "Detecting sensors fault, anomaly, and outlier in the IoT: a survey on the challenge and solution," *Electronic*, vol. 9, no. 511, pp. 1–15, 2020.

[10] A. Junaid, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A review of performances, energies, and privacies of intrusions detections system for IoTs," *Electronic*, vol. 9, no. 629, pp. 1–24, 2020.

[11] S. Bhawana, S. Lokesh, and L. Chhagan, "Anomaly detections technique using deep learnings in IoT: a survey," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 146–149, Dubai, United Arab Emirates, 2019.

[12] F. Muhammad and S. Alberto, "Anomaly detection, analysis and prediction techniques in iot environment: a systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019.

[13] K. S. Nilesh and M. Indrajit, "Machine learnings based anomaly detections for IoT Networks," in *Proceeding of the Fourth International Conferences on Trend in Electronic and Informatics (ICOEI 2020)*, pp. 787–794, Tirunelveli, India, 2020.

[14] A. Ashikin, S. Norhalina, and T. R. Y. Iwan, "Designing deep neural network with chicken swarm optimization for violence video classification using VSD2014 dataset," *Recent Advances on Soft Computing and Data Mining*, vol. 978, pp. 47–56, 2020.

[15] M. Xianbing, L. Yu, G. Xiaozhi, and Z. Hengzhen, "A new bio-inspired algorithms: chicken swarm optimizations," in *Advanced in Swarm Intelligences, ICSI*, Y. Tan, Y. Shi, and C. A. C. Coello, Eds., vol. 8794 of Lectured note in computer sciences, pp. 86–94, 2014.

[16] R. S. Kumar, R. Mohandas, and J. Christhudass, "A Brief Overview of Context Aware System," *Journal of Computational Science and Intelligent Technologies*, vol. 2, no. 2, pp. 17–23, 2021.

[17] R. Jayamma, "Improving The Performances of WSN Using Data Scheduler and Hierarchical Tree," *Journal of Computational Science and Intelligent Technologies*, vol. 2, no. 2, pp. 07–16, 2021.

[18] R. Mugesh, "A Survey on Security Risks in Internet of Things (IoT) Environment," *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 2, pp. 01–08, 2020.

[19] A. Nusaybah, A. Reem, and M. B. Seyed, "HLMCC: a hybrid learning anomaly detection model for unlabeled data in Internet of Things," *IEEE Access*, vol. 7, pp. 179492–179504, 2019.

[20] T. V. Khoa, Y. M. Saputra, D. T. Hoang et al., "Collaborative learning models for cyberattack detection system in IoT Industry 4.0," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Seoul, South Korea, 2020.

[21] H. S. Iqbal, B. A. Yoosef, A. Fawaz, and I. K. Asif, "IntruDTree: a machine learning-based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 754, pp. 1–15, 2020.