Computers, Materials & Continua
DOI: 10.32604/cmc.2022.032083
Article

Tech Science Press

Blockchain Assisted Intrusion Detection System Using Differential Flower Pollination Model

Mohammed Altaf Ahmed[1], Sara A Althubiti[2], Dronamraju Nageswara Rao[3], E. Laxmi Lydia[4], Woong Cho[5], Gyanendra Prasad Joshi[6] and Sung Won Kim[7,*]

[1]Department of Computer Engineering, College of Computer Engineering & Sciences,
Prince Sattam Bin Abdulaziz University, Al-Kharj, 11942, Saudi Arabia
[2]Department of Computer Science, College of Computer and Information Sciences, Majmaah University,
Al-Majmaah, 11952, Saudi Arabia
[3]Department of Computer Science and Engineering, Chitkara University Institute of Engineering and Technology,
Chitkara University, Punjab, 140401, India
[4]Department of Computer Science and Engineering, Vignan's Institute of Information Technology,
Visakhapatnam, 530049, India
[5]Department of Software Convergence, Daegu Catholic University, Gyeongsan, 38430, Korea
[6]Department of Computer Science and Engineering, Sejong University, Seoul, 05006, Korea
[7]Department of Information and Communication Engineering, Yeungnam University,
Gyeongsan-si, Gyeongbuk-do, 38541, Korea
*Corresponding Author: Sung Won Kim. Email: swon@yu.ac.kr

Received: 06 May 2022; Accepted: 07 June 2022

Abstract: Cyberattacks are developing gradually sophisticated, requiring effective intrusion detection systems (IDSs) for monitoring computer resources and creating reports on anomalous or suspicious actions. With the popularity of Internet of Things (IoT) technology, the security of IoT networks is developing a vital problem. Because of the huge number and varied kinds of IoT devices, it can be challenging task for protecting the IoT framework utilizing a typical IDS. The typical IDSs have their restrictions once executed to IoT networks because of resource constraints and complexity. Therefore, this paper presents a new Blockchain Assisted Intrusion Detection System using Differential Flower Pollination with Deep Learning (BAIDS-DFPDL) model in IoT Environment. The presented BAIDS-DFPDL model mainly focuses on the identification and classification of intrusions in the IoT environment. To accomplish this, the presented BAIDS-DFPDL model follows blockchain (BC) technology for effective and secure data transmission among the agents. Besides, the presented BAIDS-DFPDL model designs Differential Flower Pollination based feature selection (DFPFS) technique to elect features. Finally, sailfish optimization (SFO) with Restricted Boltzmann Machine (RBM) model is applied for effectual recognition of intrusions. The simulation results on benchmark dataset exhibit the enhanced performance of the BAIDS-DFPDL model over other models on the recognition of intrusions.

Keywords: Internet of things; feature selection; intrusion detection; blockchain; security; deep learning

1 Introduction

The Internet of Things (IoT) is an interlinked system that depends on approved protocols that interchange information amongst the gadgets functioning through the internet. Latest developments in IoT launch the ideology of smartness to gadgets, homes, sensors, cities, and streets. IoT is one of the leading and rising domains of communication technology and modern computing done great contributions to several fields, starting from agricultural field to automobile automation [1]. Today, IoT has further termed the Internet of Everything (IoE) why because it has been dealing with every single type of linked device or gadget in day-to-day routine. In 2025, the number of connected gadgets is expected to reach 21.5 billion. IoT is a compilation of numerous layers adding a network layer [2]. The network layer model is depending on the old layers of Internet transmission and is mostly accountable for sending data packets among hosts. Also, the network layer is complicated and a prone part of IoT substructure resulting in several safety threats [3]. Nonetheless, numerous security structures have been placed for addressing the security problems. Such structures need fixing in the IoT substructure and or the gadgets to function effectively resolving security menaces. Inappropriately, many security structures need substantial storage and computational power [4].

Intrusion detection systems (IDS) identify doubtful events and well-known menaces and produce warnings [5]. Intrusions can be detected as any event which makes damage to an information system. IDS can be hardware or software systems able to identify any other malevolent actions in computer systems [6]. The objective of intrusion detection systems is to observe the computer network to identify irregular conduct that might not be noticed by an orthodox packet filter. It is more important to reach a higher extent of cyber resilience towards the malevolent actions and to recognize unauthorized accessibility to a computer system by examining the network packets for signs of malevolent action [7]. During the past decades, IDSs is highly dependable and effective since invaders also enhanced many expanded assault modes for defeating such identification systems. Moreover, old IDSs could not cooperate with that the complicated network layers in IoT [8]. The recent advances in intelligent systems put forwarded the authors to use of dispersed IDSs in amalgamation with several machine learning (ML) algorithms, for instance, reinforcement learning (RL), artificial neural network (ANN), and deep learning (DL) [9]. Usual ANNs contain few restrictions in handling the difficulty of IDSs. Enlightening technology through deciding these faults is a prerequisite for understanding the promise of IDSs auspicious in the actual ecosphere [10]. The foremost contribution of this study is applying blockchain (BC) for a multi agent system and tested with a known dataset for its efficacy.

Alkadi et al. [11] present a deep BC framework (DBF) planned for offering security-based distributed ID and privacy-based BC with smart contracts from IoT networks. The ID technique was utilized by bidirectional long short-term memory (BiLSTM) DL technique for dealing with sequential network data and is measured utilizing the dataset of UNSW-NB15 and BoT-IoT. The privacy-based BC and smart contract techniques were established utilizing the Ethereum library for providing privacy to distributed ID engines. Tariq et al. [12] purpose for enhancing knowledge of the challenges for securing future digital frameworks but it can be still developed. Then presenting the structure creating big data, the functionality-based fog structure was determined. Liang et al. [13] presented the collaborative clustering-characteristic-based data fusion technique to ID from BC-based method, whereas the mathematical process of data fusion was planned and AI technique was utilized for training and analysing data clusters from BC network.

Cheema et al. [14] examine the distributed ML based IDS in IoT employing BC technology. Particularly, spectral partitioning was presented for dividing the IoT network as to autonomous systems (AS) allowing traffic observation for ID that is executed by chosen AS border region nodes from the distributed approach. The IDS was dependent upon ML, whereas an SVM technique was trained utilizing prominent IoT dataset and recognition of attackers is given. Zhang et al. [15] present the BC supported massive IoT data collection (MIDC) intelligent structure for supporting the security, trust, and efficacy of enormous data collection to large-scale heterogeneous WSN. Particularly, it can present a series of novel technology to structure. Primary, it can be structured as a Large-Scale Heterogeneous WSN Collaborative Identity Verification protocol for ensuring reliable data sources. Secondary, it can be structured as a Hierarchical Massive Data Aggregation technique for collecting huge IoT data secure and efficient. Tertiary, it can demonstrate the BC based Massive IoT Data Management technique for constructing trust amongst distinct parties.

This paper presents a new Blockchain Assisted Intrusion Detection System using Differential Flower Pollination with Deep Learning (BAIDS-DFPDL) model in IoT Environment. The presented BAIDS-DFPDL follows blockchain (BC) technology for effective and secure data transmission among the agents. Besides, the presented BAIDS-DFPDL model designs Differential Flower Pollination based feature selection (DFPFS) technique to elect features. Finally, sailfish optimization (SFO) with Restricted Boltzmann Machine (RBM) model is applied for effectual recognition of intrusions. The simulation results on benchmark dataset exhibit the enhanced performance of the BAIDS-DFPDL model over other models on the recognition of intrusions.

2 Design of BAIDS-DFPDL Model

In this study, a new BAIDS-DFPDL model has been developed for identification and classification of intrusions in the IoT environment. To accomplish this, the presented BAIDS-DFPDL model follows BC technology for effective and secure data transmission among the agents. Besides, the presented BAIDS-DFPDL model designed DFPFS technique to elect features. At last, the SFO algorithm with RBM model is applied for effectual recognition of intrusions. Fig. 1 showcases the overall process of BAIDS-DFPDL technique.

2.1 BC Enabled Secure Communication

The presented BAIDS-DFPDL model follows BC technology for effective and secure data transmission among the agents. BC transformed the architecture of the transactional system from a star to a point to point (P2P) design. This architecture enables two parties to handle one another straightaway through security and encryption based on algorithm and code privacy [16]. Meanwhile, the party engaged in the transactional network are essential to entrust the used approach to establish mutual trust, there is no need for knowledge regarding the dependability of party. Here, private chain is utilized for securing interaction among agents. The scheme incorporates the embedded dataset and the BC node in a similar agent, stopping and starting simultaneously. The modern interaction would be included in the cache region, decreasing the searching time, and speeding up the use and disk storage dataset through agents. Saving a copy of databases in each agent would decrease the overhead of transmission agents by implementing a local search of the recently upgraded dataset.

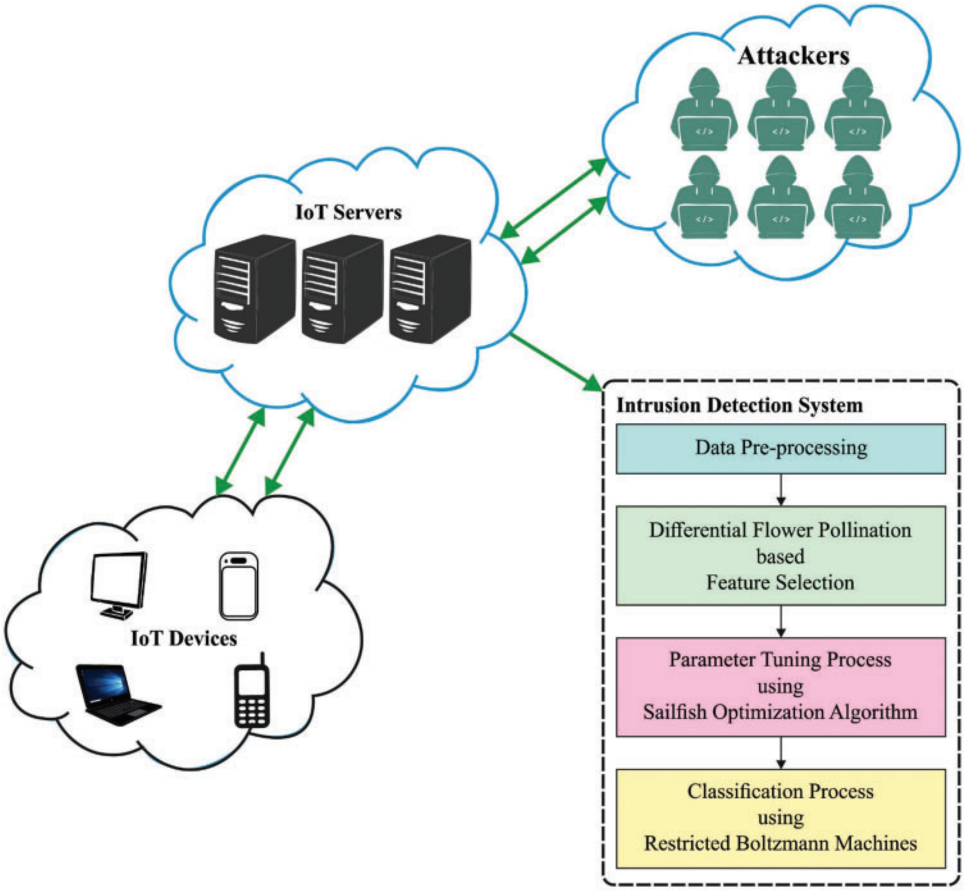
Attackers
IoT Servers
IoT Devices
Intrusion Detection System
Data Pre-processing
Differential Flower Pollination based Feature Selection
Parameter Tuning Process using Sailfish Optimization Algorithm
Classification Process using Restricted Boltzmann Machines

Figure 1: Overall process of BAIDS-DFPDL technique

The transmission agent of the analysis and detection model is a great node for the BC model. Fig. 2 depicts the framework of BC. It comprises the whole BC ledger and is utilized for guaranteeing each node in the system could get the right copy and connected with one another. Each transmission agent would be complete node. They distribute and contain the whole BC ledger and are necessary to validate each transmission record in the BC. Another agent in all the modules is empowered as light node and connected with the transmission agents which are parental nodes. Such light node contains block header of the block and is employed for checking whether the parental nodes get interfered. The BC model is applied for regulating and validating behavior of transmission agents and to guarantee the security of the network, along with construct reliance in IoT network whereby several parties are included that have their own agent. Additionally, information from transmission messages is employed for analyzing attacks and improving the agent by employing reinforcement learning.

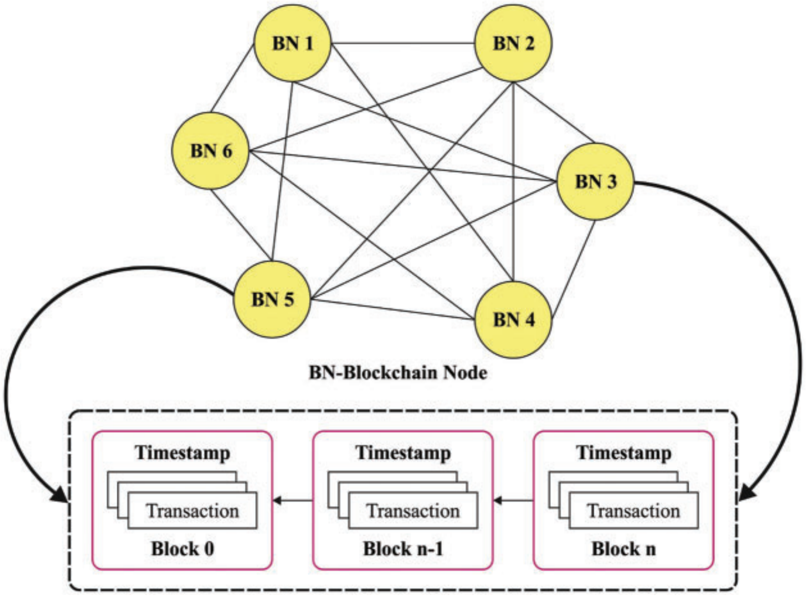

Figure 2: Structure of blockchain

2.2 Process Involved in DFPFS Model

At this stage, the DFPFS technique has been developed for the effectual choice of features. DFP is a fusion of two separate Meta-heuristic approaches such as flower pollination algorithm (FPA) and differential evolution (DE). DE is a remarkably successful variant of evolutionary algorithm that includes important evolutionary processes comprising recombination and mutation [17]. In the meantime, FPA is dominant in searching space exploration because of its usage of Lévy flight tasks. The fusion of FPA and DE conserves the benefit of the self-adoptive mutation of DE and considerably improves the efficacy of the contraction performance of DE transitional arithmetical reintegration. Thus, the integrated DFP approach has an effective exploitation ability and has an escape strategy to avoid early convergences. According to the arbitrarily made population of candidate solutions, the DFP implements global and local searching tasks. The rate of two search processes are depending upon a possibility p based on the subsequent rules:

When $r < p$ then

{Execute global searching task}

Otherwise

{Execute local searching task},

Here r indicates an arbitrary value lies in the interval of [0, 1]. p is usually fixed as 0.8 based on.

The global and local searching tasks are defined in the following:

(i) The FPA-based global searching function is represented as follows

$$X_i^{trial} = X_i^g + L.\left(X_i^g - X_{best}\right), \tag{1}$$

In Eq. (1), g indicates the present generation. X_i^{trial} denotes a recently made test solution. L signifies the amount made from the Lévy distribution.

(ii) The DE-based local searching function is represented as

$$X_{i,g}^{mutated} = x_{r1,g} + F.\left(x_{r2,g} - x_{r3,g}\right), \tag{2}$$

In Eq. (2), $1, r2$, and $r3$ denote three arbitrary values. F indicates a mutation scaling factor. The variable F is made from a Gaussian distribution using the standard deviation $= 0.15$ and the mean $= 0.5$.

$$X_{j,i,g+1}^{crossed} = \begin{cases} X_{j,i}^{mutated}, & if\ r\ and_j \leq Cr\ or\ j = rnb\,(i)\,, \\ X_{j,i,g}, & if\ r\ and_j > Cr\ and\ j \neq rnb\,(i)\,, \end{cases} \tag{3}$$

In Eq. (3), Cr indicate the crossover possibility that is usually fixed as 0.8.

It is stated that the variable L is an arbitrary amount produced from the Lévy distribution. The mathematical expression of this distribution can be given below:

$$L \sim \frac{\lambda \Gamma\,(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi s^{1+\lambda}}, \tag{4}$$

Here $s = Ul|V|^{1/\lambda}$ and U and $V \sim$ standard $(0, 1)$. $\Gamma\,(\lambda)$ indicates the Gamma distribution using $\lambda = 3/2$.

The fitness function of the DFPFS technique assumes the classifier accuracy and the quantity of features which is selected. It optimizes the classifier accuracy and reduces the set size of the selected features. Thus, the succeeding fitness function is utilized for evaluating separate solutions, as displayed in Eq. (5).

$$Fitness = \alpha * \ ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{5}$$

Whereas ErrorRate denotes the classification fault rate utilizing the selected features. ErrorRate is computed as the percentage of inaccurate classified to the number of classifiers made, denoted as a value between 0 and 1. (ErrorRate is the supplement of the classifier accuracy), $\#SF$ is considered as the number of selected features and $\#All\_F$ is regarded as the aggregate sum of attributes in the novel dataset. α is utilized for controlling the significance of classifier quality and subset length. In our experiments, α is fixed to 0.9.

2.3 Optimal RBM Based Classification

Next to the election of features, the RBM model is employed to detect and classify intrusions. RBM is known as Stochastic Neural Network since they assume a probabilistic method. They are comprised of:

- Single visible unit or input layer
- Single hidden unit or hidden layer
- Bias unit

RBM is applied as a generated method for labeled or unlabeled datasets. There is no intra-layer interaction in RBM that is the constraint [18]. In the initial node of hidden layer, an input value is increased with weight and included in bias and the outcome is transmitted to the activation function and later the output is generated. The energy of joint formation of visible and hidden layers is evaluated by:

$$E\left(v,\ h\right) = -\sum_{i} a_i v_i - \sum_{j} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{6}$$

v_i and h_j indicates the binary state of i and j visible and hidden units, $a_i; b_j$ indicates the biases and w_{ij} indicates the weight among themselves. For calculating the involvement of the visible or hidden units and biases, the dot product is evaluated. Then, the possibility of each probable pair of visible and hidden vectors is evaluated by the energy function as:

$$p\left(v,\ h\right) = \frac{1}{Z} e^{-E(v,h)} \tag{7}$$

From the equation, Z indicates the partition function:

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{8}$$

The likelihood that the network allocates to a visible unit is evaluated by:

$$p\left(v\right) = \frac{1}{Z} \sum_{h} e^{-E(v,h)} \tag{9}$$

The derivation of log likelihood of a trained vector in terms of weight is evaluated.

$$\frac{\partial \log p\left(v\right)}{\partial w_{ij}} = \left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{model} \tag{10}$$

That results in a simple rule with $\in$ *as* learning rate:

$$\Delta w_{ij} = \in \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}\right) \tag{11}$$

Geoffrey Hinton presented a fast learning method where recreation is formed by setting v_i to 1:

$$\Delta w_{ij} = \varepsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{reconstruction}\right) \tag{12}$$

This model has the uppermost trained time amongst each model, consequently, the general trained time of module is equivalent to the trained time of the SRBMM model. The values of network traffic feature in data corresponding to "visible" unit of the RBM because they are observed. The pre-training stage involves a stack of RBM with contrast divergence approach. The extracted feature from initial demonstration RBM is utilized to train in the following RBM and lastly, back-propagation of error can be implemented. In such cases, it is impossible to utilize Eq. (7) in this technique. To substitute visible unit with linear unit, Gaussian RBM consider there are real-value visible layers amongst [0, 1]:

$$E\left(v,\ h\right) = \sum_{i} \frac{\left(v_i - a_i\right)^2}{2\sigma_i^2} - \sum_{j} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \tag{13}$$

2.4 Parameter Optimization Using SFO Algorithm

For tuning the parameters [19–21] related to the RBM model, the SFO algorithm is utilized. SFO [22] is a population based metaheuristic approach that is motivated by the attack-alternated method of collection of hunting sailfish that hunt a school of sardine. This hunting method offers dominance for hunters by providing the opportunity of soring the energy. It considers that: sailfish along with sardine populations. The sailfish was considered the solution candidate and the problem parameter are the position of sailfish from the searching region. This algorithm tries to randomize the effort of searching agents (sardine and sailfish) highly possible. The sailfish is considered as disseminated from the searching space, however, the position of sardines is used to find an optimal solution from the searching space.

The sailfish with optimal fitness values are called 'elite' sailfish and the position in i^{th} iterations are presented by $P^i_{SlfBest}$. For sardine, the 'injured' is one with optimal fitness values and the position in i^{th} iterations are represented by $P^i_{SrdInjured}$. For each iteration, the sardine and sailfish places are upgraded. In $i + 1^{th}$ iteration, an original location P^{i+1}_{Slf} of sailfish has upgraded with 'elite' sailfish and 'injured' sardine as follows.

$$P^{i+1}_{Slf} = P^i_{SlfBest} - \mu_i \times \left(rnd \times \frac{P^i_{SlfBest} + P^i_{SrdInjured}}{2} - P^i_{Slf} \right) \tag{14}$$

In Eq. (14), P^i_{Slf} indicates prior location of Slf^{th} sailfish and rnd indicate the random value between zero and one and μ_i denotes the coefficient that is generated by using the following equation.

$$\mu_i = 2 \times rnd \times PrD - PrD \tag{15}$$

In Eq. (15) PrD indicates the prey density which indicates the prey amount in each iteration. From each iteration, the values of PrD, calculated in the following expression, minimize when the prey amount is minimized in group hunting.

$$PrD = 1 - \frac{Num_{Slf}}{Num_{Slf} + Num_{Srd}} \tag{16}$$

In Eq. (16), Num_{Slf} and Num_{Srd} are the amount of sailfishes as well as sardines correspondingly.

$$Num_{Slf} = Num_{Srd} \times \ Prcnt \tag{17}$$

In Eq. (17), $Prcnt$ indicates the percentage of sardine population that generated the most important sailfish population. The main quantity of sardines is considered greater than the sailfish amount. The sardine position was upgraded in every iteration as follows.

$$P^{i+1}_{Srd} = rnd\,(0,\ 1) \times \left(P^i_{SLfBest} - P^i_{Srd} + ATK\right) \tag{18}$$

$$ATK = A \times (1 - (2 \times itr \times \kappa)) \tag{19}$$

From the above equation, P^i_{Srd} and P^{i+1}_{Srd} correspondingly indicates the prior and upgraded positions of sardine and ATK indicates the sailfish attack strength in itr iteration. Currently, the quantity of sardines upgrades the position and the displacement amount depends on ATK. In order to decrease the ATK assist the convergences of searching agent. With the variable ATK, the sardine amount upgrades the position (Γ) and the variable amount of (δ) is evaluated by:

$$\Gamma = Num_{Srd} \times ATK \tag{20}$$

$$\delta = v \times ATK \tag{21}$$

In the equation, v indicates the variable amount and Num_{Srd} indicates the sardine amount. Once the sardine develops proper into sailfish, the sailfish upgrade the location following the sardine, and the sardine was detached from the population. The selection of sardine and sailfish randomly guarantees the exploration of searching space. Therefore, the attack strength of sailfish is minimized, each iteration gives an opportunity for sardine to escape from optimal sailfish which assists in exploitation. The ATK variable tries to discover the balance between exploration and exploitation.

3 Performance Validation

In this section, the experimental validation of the BAIDS-DFPDL model is tested using Bot-IoT dataset and NSL-KDD dataset.

The first Bot-IoT dataset comprises distinct classes such as Normal (8945), DDoS (2766), DoS (6391), Reconnaissance (298), Keylogging (73). Then, the NSL-KDD dataset holds several class labels namely Dos (45927), Probe (11656), R2L (995), U2R (52), and Normal (67343).

Tab. 1 and Fig. 3 report a brief classification outcome of the BAIDS-DFPDL model under distinct batch size (BS) and epoch count on Bot-IoT dataset. The results implied that the BAIDS-DFPDL model has accomplished enhanced classifier results in all aspects. For instance, with BS of 32, the BAIDS-DFPDL model has gained improved classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.28%, 99.22%, 99.25%, 99.19%, and 98.98% respectively. In addition, with BS of 64, the BAIDS-DFPDL methodology has attained higher classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.02%, 99.03%, 99.12%, 99.21%, and 99.31% correspondingly. Along with that, with BS of 128, the BAIDS-DFPDL approach has reached improved classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.22%, 99.20%, 99.23%, 99.27%, and 99.17% correspondingly.

Table 1: Result analysis of BAIDS-DFPDL technique with various measures under Bot-IoT dataset

No. of epochs	Accuracy	Precision	Recall	F-score	MCC
Batch-size = 32					
Epoch-50	99.44	99.50	99.56	99.27	99.38
Epoch-100	99.23	98.93	99.03	99.15	99.06
Epoch-150	99.02	99.40	99.38	99.56	98.79
Epoch-200	99.16	99.30	99.21	99.34	98.71
Epoch-250	99.44	99.14	99.02	98.80	98.91
Epoch-300	99.40	99.02	99.29	99.00	99.03
Average	99.28	99.22	99.25	99.19	98.98
Batch-size = 64					
Epoch-50	99.02	98.94	99.50	99.28	99.35
Epoch-100	99.14	98.88	99.31	99.25	99.15
Epoch-150	99.19	99.40	99.38	99.58	99.43
Epoch-200	98.87	98.75	98.77	98.80	99.38
Epoch-250	98.74	99.35	98.70	99.03	99.13
Epoch-300	99.13	98.83	99.06	99.34	99.43
Average	99.02	99.03	99.12	99.21	99.31
Batch-size = 128					
Epoch-50	99.60	98.86	99.55	99.33	98.78
Epoch-100	99.43	99.33	98.99	99.44	99.16
Epoch-150	98.71	99.31	99.59	99.33	99.28
Epoch-200	99.30	99.22	99.12	98.98	99.52

(Continued)

Table 1: Continued

No. of epochs	Accuracy	Precision	Recall	F-score	MCC
Epoch-250	98.72	99.20	98.78	99.09	99.53
Epoch-300	99.54	99.30	99.33	99.45	98.74
Average	99.22	99.20	99.23	99.27	99.17

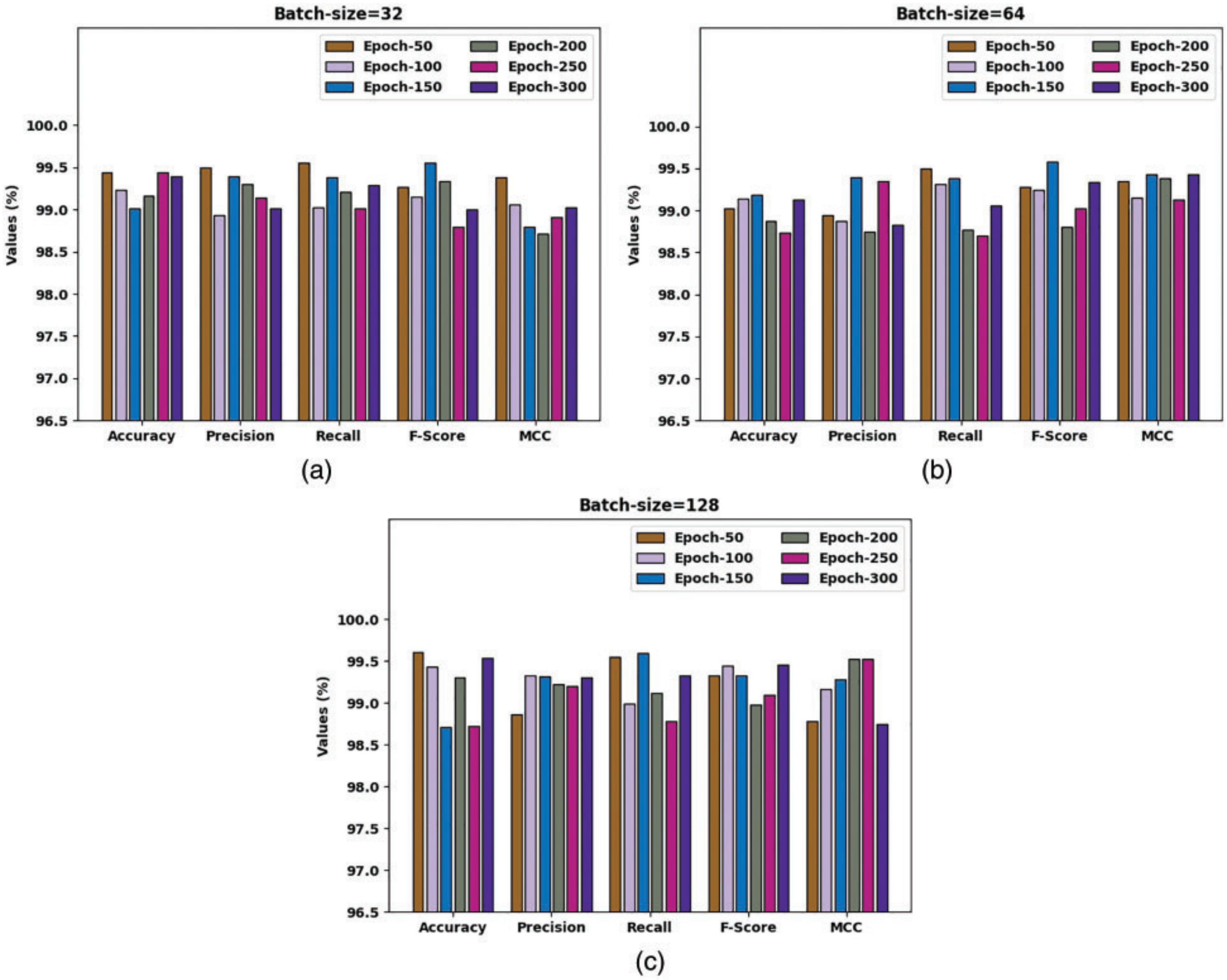

Figure 3: Result analysis of BAIDS-DFPDL technique under Bot-IoT dataset (a) BS = 32, (b) BS = 64, and (c) BS = 128

A brief precision-recall examination of the BAIDS-DFPDL model on Bot-IoT dataset is portrayed in Fig. 4. By observing the figure, it is noticed that the BAIDS-DFPDL approach has accomplished maximal precision-recall performance under all classes.

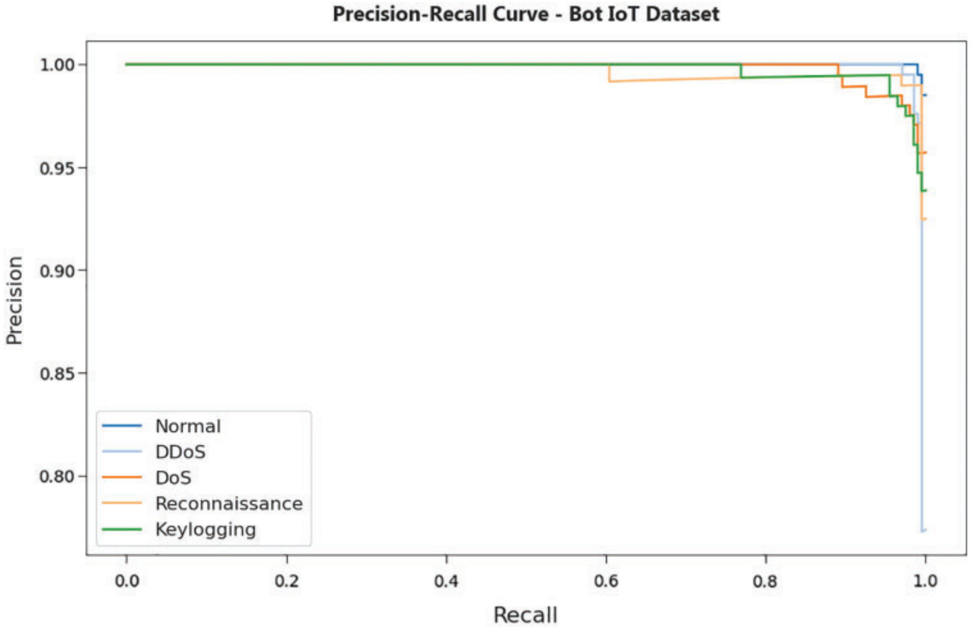

Figure 4: Precision-recall curve analysis of BAIDS-DFPDL approach under Bot-IoT dataset

A detailed ROC investigation of the BAIDS-DFPDL model on Bot-IoT dataset is portrayed in Fig. 5. The results exposed that the BAIDS-DFPDL model has exhibited its ability in categorizing five different classes on the test dataset.

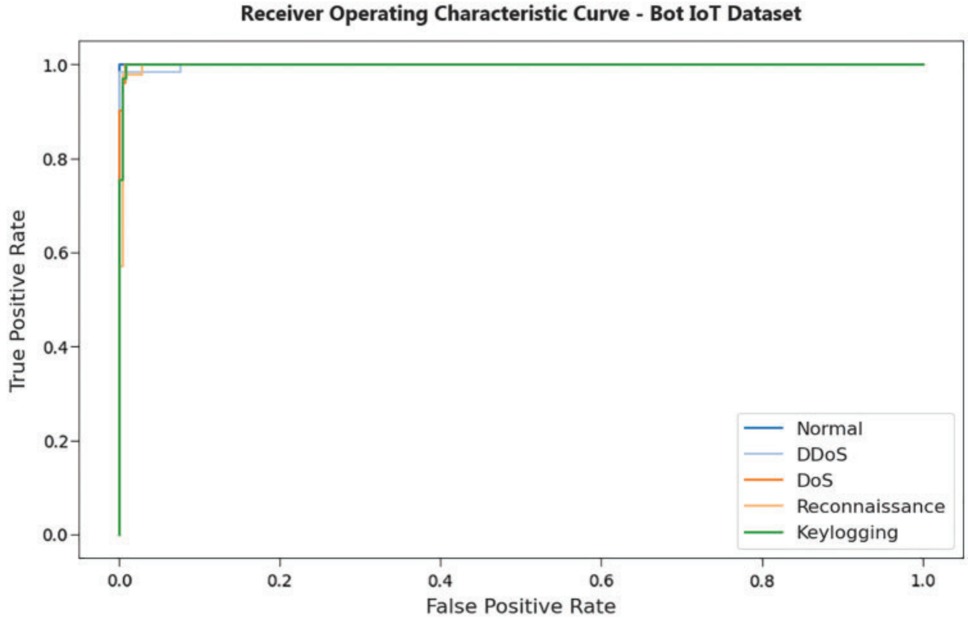

Figure 5: ROC curve analysis of BAIDS-DFPDL approach under Bot-IoT dataset

Tab. 2 and Fig. 6 report a brief comparative study of the BAIDS-DFPDL model on the test Bot-IoT dataset [23,24]. The experimental values implied that the CART model has shown worse performance with least $accu_y$, $prec_n$, and $reca_l$ of 80.33%, 79.86%, and 79.66% respectively. At the same time, the MLP, KNN, NB, and SVM models have reached slightly improved performance over other models. Then, the RF and C4.5 models have tried to accomplish reasonable outcomes. Though the SIDS model has resulted in considerable classification results with $accu_y$, $prec_n$, and $reca_l$ of 98.92%, 97.51%, and 99.13%, the presented BAIDS-DFPDL model has depicted maximum performance $accu_y$, $prec_n$, and $reca_l$ of 99.28%, 99.22%, and 99.25% respectively.

Table 2: Comparative analysis of BAIDS-DFPDL technique with existing algorithms under Bot-IoT dataset

Methods	Accuracy	Precision	Recall
C4.5 algorithm	91.73	91.97	91.70
NB algorithm	87.71	87.04	87.78
RF algorithm	91.70	90.62	91.78
MLP algorithm	86.46	85.22	85.86
SVM algorithm	88.77	87.78	87.40
CART algorithm	80.33	79.86	79.66
KNN algorithm	87.66	86.20	86.74
SIDS model	98.92	97.51	99.13
BAIDS-DFPDL	99.28	99.22	99.25

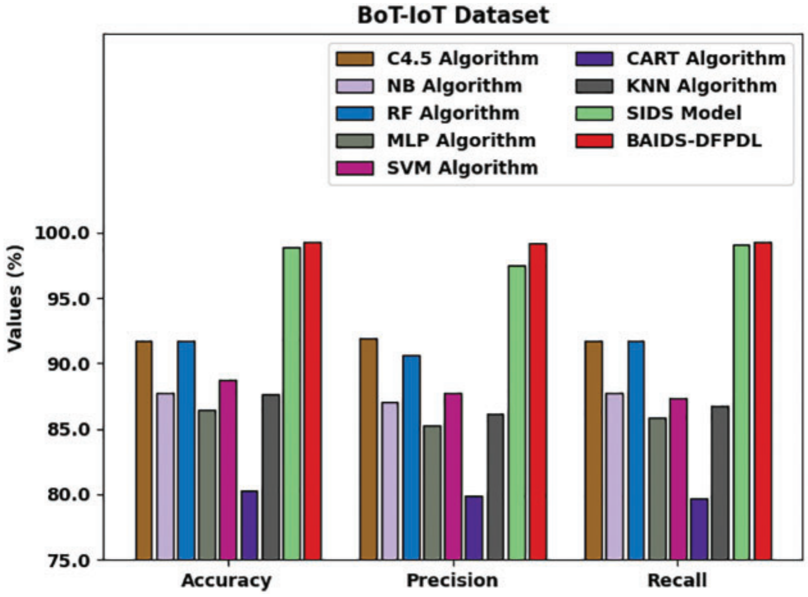

Figure 6: Comparative analysis of BAIDS-DFPDL technique under Bot-IoT dataset

Tab. 3 and Fig. 7 illustrate a brief classification outcome of the BAIDS-DFPDL algorithm under distinct BS and epoch counts on NSL-KDD dataset. The outcomes exposed that the BAIDS-DFPDL model has accomplished enhanced classifier outcomes in all aspects. For instance, with BS of 32, the BAIDS-DFPDL algorithm has gained increased classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 90.26%, 89.98%, 90.56%, 89.87%, and 89.56% correspondingly. Furthermore, with BS of 64, the BAIDS-DFPDL model has gained maximal classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 90.46%, 88.90%, 90.93%, 90.10%, and 90.16% respectively. Eventually, with BS of 128, the BAIDS-DFPDL algorithm has gained enhanced classification performance with average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 91.01%, 90.79%, 90.78%, 89.72%, and 90.35% correspondingly.

Table 3: Result analysis of BAIDS-DFPDL technique with various measures under NSL-KDD dataset

No. of epochs	Accuracy	Precision	Recall	F-score	MCC
Batch-size = 32					
Epoch-50	92.07	88.16	88.62	89.76	92.45
Epoch-100	92.35	88.60	90.55	90.32	90.02
Epoch-150	88.71	92.51	92.02	88.51	89.51
Epoch-200	90.02	89.83	92.50	91.59	88.65
Epoch-250	87.81	88.56	91.80	88.12	88.78
Epoch-300	90.62	92.24	87.88	90.92	87.97
Average	90.26	89.98	90.56	89.87	89.56
Batch-size = 64					
Epoch-50	88.41	91.40	91.93	89.11	92.30
Epoch-100	91.12	88.05	91.68	91.86	89.43
Epoch-150	92.11	89.90	88.00	91.82	90.43
Epoch-200	90.93	88.40	91.97	88.28	89.22
Epoch-250	91.26	87.91	90.47	89.28	89.02
Epoch-300	88.90	87.72	91.55	90.24	90.55
Average	90.46	88.90	90.93	90.10	90.16
Batch-size = 128					
Epoch-50	91.44	88.09	89.94	92.20	89.73
Epoch-100	92.08	90.87	91.31	89.00	90.62
Epoch-150	91.10	91.63	91.61	89.44	90.01
Epoch-200	90.44	92.24	90.51	89.96	88.21
Epoch-250	90.98	90.49	89.57	87.92	92.44
Epoch-300	90.21	91.40	91.74	89.82	91.11
Average	91.04	90.79	90.78	89.72	90.35

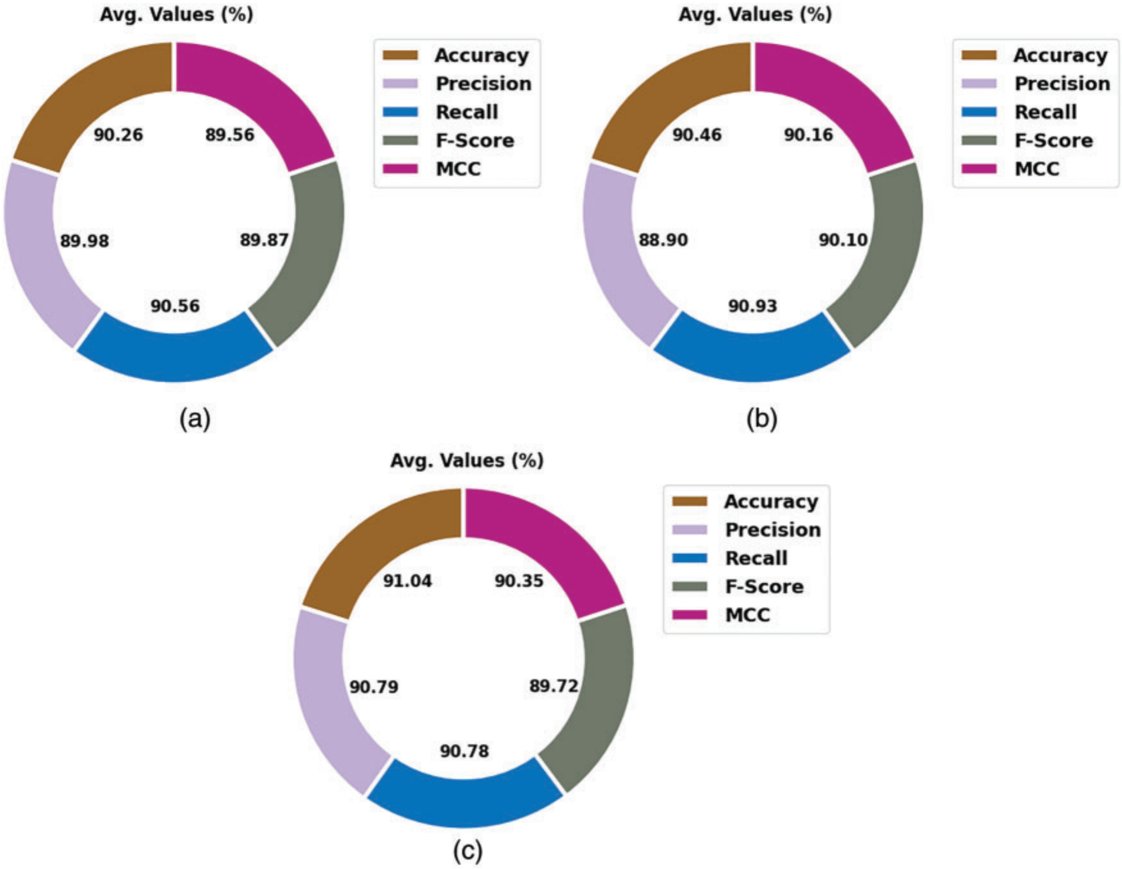

Figure 7: Average analysis of BAIDS-DFPDL technique under NSL-KDD dataset (a) BS = 32, (b) BS = 64, and (c) BS = 128

Tab. 4 and Fig. 8 demonstrate a brief comparative analysis of the BAIDS-DFPDL technique on the test NSL-KDD dataset. The experimental values exposed that the SVM approach has demonstrated least performance with minimal $accu_y$, $prec_n$, and $reca_l$ of 70.32%, 70.84%, and 70.37% correspondingly. Besides, the MLP, KNN, NB, and CART algorithms have reached somewhat enhanced performance over other methodologies.

Table 4: Comparative analysis of BAIDS-DFPDL technique with existing algorithms under NSL-KDD dataset

Methods	Accuracy	Precision	Recall
C4.5 algorithm	80.65	80.86	81.82
NB algorithm	76.53	78.02	76.68
RF algorithm	81.19	81.08	82.27
MLP algorithm	78.21	78.02	79.77
SVM algorithm	70.32	70.84	70.37
CART algorithm	81.22	81.30	82.79

(Continued)

Table 4: Continued

Methods	Accuracy	Precision	Recall
KNN algorithm	80.34	80.18	80.84
HIDS model	83.60	85.13	83.56
BAIDS-DFPDL	91.04	90.79	90.78

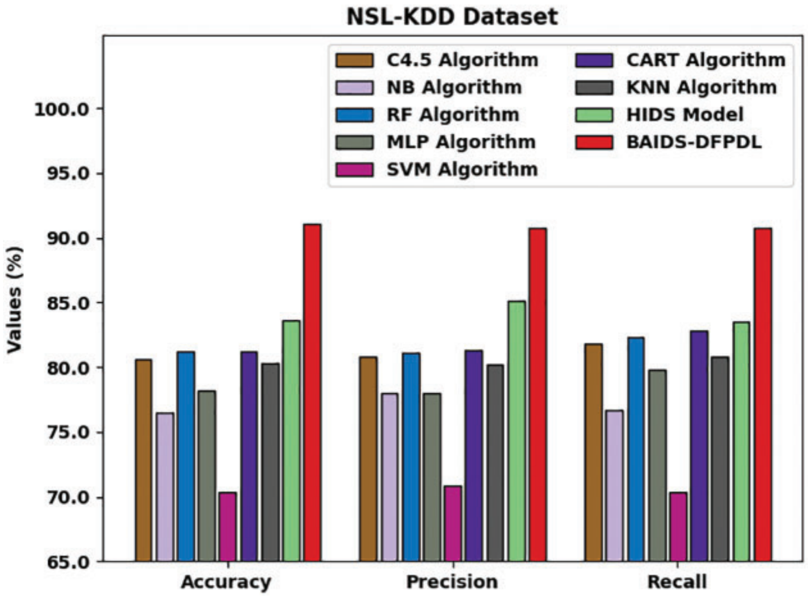

Figure 8: Comparative analysis of BAIDS-DFPDL technique under NSL-KDD dataset

Followed by, the RF and C4.5 models have tried to accomplish reasonable outcomes. But, the HIDS algorithm has resulted in considerable classification results with $accu_y$, $prec_n$, and $reca_l$ of 83.60%, 85.13%, and 83.56%, the presented BAIDS-DFPDL methodology has exhibited higher performance $accu_y$, $prec_n$, and $reca_l$ of 91.04%, 90.79%, and 90.78% correspondingly. Therefore, the proposed BAIDS-DFPDL model has showcased excellent performance over all the other models.

4 Conclusion

In this study, a new BAIDS-DFPDL model has been developed for identification and classification of intrusions in the IoT environment. To accomplish this, the presented BAIDS-DFPDL model follows BC technology for effective and secure data transmission among the agents. Besides, the presented BAIDS-DFPDL model designed DFPFS technique to elect features. At last, the SFO algorithm with RBM model is applied for effectual recognition of intrusions. The simulation results on benchmark dataset exhibit the enhanced performance of the BAIDS-DFPDL model over other models in the recognition of intrusions in the IoT environment. Thus, the BAIDS-DFPDL technique can be exploited for accomplishing security in the IoT environment. In future, the performance of the BAIDS-DFPDL model can be boosted by the design of feature reduction approaches.

Funding Statement: This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A1A03039493), in part by the NRF grant funded by the Korea government (MSIT) (NRF-2022R1A2C1004401), and in part by the 2022 Yeungnam University Research Grant.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. A. Khan, M. M. Khan, K. M. Khan, J. Arshad and F. Ahmad, "A blockchain-based decentralized machine learning framework for collaborative intrusion detection within UAVs," *Computer Networks*, vol. 196, pp. 108217, 2021.

[2] W. Meng, W. Li, S. Tug and J. Tan, "Towards blockchain-enabled single character frequency-based exclusive signature matching in IoT-assisted smart cities," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 268–277, 2020.

[3] C. Liang, B. Shanmugam, S. Azam, A. Karim, A. Islam *et al.,* "Intrusion detection system for the internet of things based on blockchain and multi-agent systems," *Electronics*, vol. 9, no. 7, pp. 1120, 2020.

[4] V. Kelli, P. Sarigiannidis, V. Argyriou, T. Lagkas and V. Vitsas, "A cyber resilience framework for NG-IoT healthcare using machine learning and blockchain," in *ICC 2021-IEEE Int. Conf. on Communications*, Montreal, QC, Canada, pp. 1–6, 2021.

[5] D. Li, Z. Cai, L. Deng, X. Yao and H. H. Wang, "Information security model of block chain based on intrusion sensing in the IoT environment," *Cluster Computing*, vol. 22, no. S1, pp. 451–468, 2019.

[6] A. M. Krishna and A. K. Tyagi, "Intrusion detection in intelligent transportation system and its applications using blockchain technology," in *2020 Int. Conf. on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Vellore, India, pp. 1–8, 2020.

[7] H. Benaddi and K. Ibrahimi, "A review: Collaborative intrusion detection for IoT integrating the blockchain technologies," in *2020 8th Int. Conf. on Wireless Networks and Mobile Communications (WINCOM)*, Reims, France, pp. 1–6, 2020.

[8] V. Hemamalini, G. Zayaraz and V. Vijayalakshmi, "BSPC: Blockchain-aided secure process control for improving the efficiency of industrial internet of things," *Journal of Ambient Intelligence and Humanized Computing*, 2022. https://link.springer.com/article/10.1007/s12652-022-03721-3.

[9] M. A. Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam *et al.,* "Federated intrusion detection in blockchain-based smart transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2523–2537, 2022.

[10] W. Li, Y. Wang, W. Meng, J. Li and C. Su, "BlockCSDN: Towards blockchain-based collaborative intrusion detection in software defined networking," *IEICE Transactions on Information and Systems*, vol. E105.D, no. 2, pp. 272–279, 2022.

[11] O. Alkadi, N. Moustafa, B. Turnbull and K. K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021.

[12] N. Tariq, M. Asim, F. A. Obeidat, M. Z. Farooqi, T. Baker *et al.,* "The security of big data in fog-enabled IoT applications including blockchain: A survey," *Sensors*, vol. 19, no. 8, pp. 1788, 2019.

[13] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He *et al.,* "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet Things Journal*, pp. 1–1, 2021. https://ieeexplore.ieee.org/abstract/document/9333661.

[14] M. A. Cheema, H. K. Qureshi, C. Chrysostomou and M. Lestas, "Utilizing blockchain for distributed machine learning based intrusion detection in internet of things," in *2020 16th Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, CA, USA, pp. 429–435, 2020.

[15] L. Zhang, F. Li, P. Wang, R. Su and Z. Chi, "A blockchain-assisted massive IoT data collection intelligent framework," *IEEE Internet Things Journal*, pp. 1, 2021. https://ieeexplore.ieee.org/abstract/document/9316679.

[16] Z. Li, A. Barenji and G. Huang, "Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 133–144, 2018.

[17] H. M. Dubey, M. Pandit and B. K. Panigrahi, "Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch," *Renewable Energy*, vol. 83, pp. 188–202, 2015.

[18] M. Wong, B. Farooq and G. -A. Bilodeau, "Discriminative conditional restricted boltzmann machine for discrete choice and latent variable modelling," *Journal of Choice Modelling*, vol. 29, pp. 152–168, 2018.

[19] O. A. Alzubi, J. A. Alzubi, K. Shankar and D. Gupta, "Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 12, pp. 1–14, 2021.

[20] G. N. Nguyen, N. H. L. Viet, M. Elhoseny, K. Shankar, B. B. Gupta *et al.,* "Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with ResNet model," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 150–160, 2021.

[21] G. P. Joshi, E. Perumal, K. Shankar, U. Tariq, T. Ahmad *et al.,* "Toward blockchain-enabled privacy-preserving data transmission in cluster-based vehicular networks," *Electronics*, vol. 9, no. 9, pp. 1–15, 2020.

[22] S. Shadravan, H. R. Naji and V. K. Bardsiri, "The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 20–34, 2019.

[23] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," *Electronics*, vol. 8, no. 11, pp. 1210, 2019.

[24] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine," *Electronics*, vol. 9, no. 1, pp. 173, 2020.