Computer Systems Science & Engineering
DOI:10.32604/csse.2022.023321
Article

Tech Science Press

Intrusion Detection System for Big Data Analytics in IoT Environment

M. Anuradha[1,*], G. Mani[2], T. Shanthi[3], N. R. Nagarajan[4], P. Suresh[5] and C. Bharatiraja[6]

[1]Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, 600119, India
[2]Department of Computer Science and Engineering, University College of Engineering Arni, Thatchur, 632326, India
[3]Department of Electronics & Communication Engineering, Kings College of Engineering, Pudukkottai, 613303, India
[4]Department of Electronics & Communication Engineering, K.Ramakrishnan College of Engineering, Tiruchirapalli, 621112, India
[5]Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, 641407, India
[6]Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Chennai, 603203, India
*Corresponding Author: M. Anuradha. Email: anuradham@stjosephs.ac.in
Received: 03 September 2021; Accepted: 29 October 2021

Abstract: In the digital area, Internet of Things (IoT) and connected objects generate a huge quantity of data traffic which feeds big data analytic models to discover hidden patterns and detect abnormal traffic. Though IoT networks are popular and widely employed in real world applications, security in IoT networks remains a challenging problem. Conventional intrusion detection systems (IDS) cannot be employed in IoT networks owing to the limitations in resources and complexity. Therefore, this paper concentrates on the design of intelligent metaheuristic optimization based feature selection with deep learning (IMFSDL) based classification model, called IMFSDL-IDS for IoT networks. The proposed IMFSDL-IDS model involves data collection as the primary process utilizing the IoT devices and is preprocessed in two stages: data transformation and data normalization. To manage big data, Hadoop ecosystem is employed. Besides, the IMFSDL-IDS model includes a hill climbing with moth flame optimization (HCMFO) for feature subset selection to reduce the complexity and increase the overall detection efficiency. Moreover, the beetle antenna search (BAS) with variational autoencoder (VAE), called BAS-VAE technique is applied for the detection of intrusions in the feature reduced data. The BAS algorithm is integrated into the VAE to properly tune the parameters involved in it and thereby raises the classification performance. To validate the intrusion detection performance of the IMFSDL-IDS system, a set of experimentations were carried out on the standard IDS dataset and the results are investigated under distinct aspects. The resultant experimental values pointed out the betterment of the IMFSDL-IDS model over the compared models with the maximum accuracy 95.25% and 97.39% on the applied NSL-KDD and UNSW-NB15 dataset correspondingly.

Keywords: Big data; cybersecurity; IoT networks; intrusion detection; deep learning; metaheuristics; intelligent systems

1 Introduction

Internet of Things (IoT) is assumed as an interconnected system depending upon authorized protocol that interchange the data [1] between the operating devices by the internet. Current developments in IoT presents the idea of providing smartness to sensors, devices, streets, cities, and homes. The IoT is one of the major and developing areas of advanced computing and transmission technique and has created significant contribution in several fields, from agriculture sectors to vehicle automation [2]. Currently, IoT is also denoted as the Internet of Everything (IoE) as it handling many kinds of day-to-day. It is predicted that in 2025 the amount of device connection might attain 21.5 billons. IoT is a composite [3] of various layers involving a network layer. Furthermore, the network layers are difficult and sensitive part of an IoT framework result in several challenges.

However, various security architectures are in place for addressing the privacy problems [4]. These architectures need installation in the IoT framework or device to function efficiently for resolving safety threats. Unfortunately, most of the security architectures need significant computation storage and power [5]. But, several methods like authentication mechanisms, light weighted encryptions are utilized for overcoming the limitations. The huge amount of nodes, such as devices or hosts are linked to the IoT, which is a major factor for the safety problems like security breach that occurred in a single node might result in failing of the entire system. To conflict with the attacks on IoT gadgets, firewall is assumed as an initial line of protection however it isn't an efficient result because of the complexity and variability of IoT frameworks.

Recently, Intrusion Detection Systems (IDSs) have become popular because of their strength. [6] initially presented the idea of an IDS in 1980 and also introduced an explanation. The concept of IDS is to determine intruders in any field. In an IoT platform, the intruder could be a host, which tries for accessing several nodes without approval. An IDS consists of 3 major features: analysis engine, response module, and agent. The agent is only in charge for collecting data from the information stream by observing actions. In previous years, the IDS have developed more efficient and reliable, however, the hacker has also established more differentiated attack methods for defeating this identification system. Additionally, conventional IDS could not manage the composite network layer in IoT [7]. Recent advancements in smart systems have stimulated scientists for employing distributed IDS in association with several Machine Learning (ML) methods like reinforcement learning (RL), deep learning (DL), and Artificial Neural networks (ANN). The regular ANN method has certain constraints in handling the difficulty of IDS [8–10]. Enhancing techniques by resolving these limitations is a demand for understanding the possibilities of IDS in real-time. The major involvement of this method is to employ blockchain for a multi-agent scheme and tested it with familiar datasets for its efficiency.

1.1 Purpose and Contribution of the Study

Researches illustrate that the highest attacks have occurred at the transport /network layers. The primary goal of this research is to examine the chance for applying the idea of DL models for IDS in IoT networks. The main objective of this study is to establish a result, creating a smart IDS that can identify the intruder and avoid attacks in IoT platforms. To attain this purpose, initially, the existing DL based IDS methods are studied. For identifying the significant privacy and security problems regarding IoT system attacks, a review has been made previously. Additionally, the limitations of present IoT gadgets are estimated. Lastly, a solution is designed by the use of DL model for combat, unexpected attacks are occurred from illegal intruders, over transport and network layers. For improving the efficiency of IDS, several optimization methods have been applied. Therefore, the paper contribution can be summarized as follows. This paper proposes an intelligent metaheuristic optimization based feature selection (FS) with deep learning (DL) based classification model, called IMFSDL-IDS for IoT networks. The proposed IMFSDL-IDS model involves data collection at the initial stage by the use of IoT devices and data preprocessing is performed. For managing big data, Hadoop ecosystem is employed. In addition, the IMFSDL-IDS technique includes a hill climbing with moth flame optimization (HCMFO) for feature subset selection to optimally elect the subset of features. Moreover, the beetle antenna search (BAS) with variational autoencoder (VAE), called BAS-VAE method is applied for the detection of intrusions. For computing the effectual intrusion detection results of the IMFSDL-IDS technique, a sequence of experimentations take place and the results are inspected with respect to several dimensions.

1.2 Organization of the Study

The residual section of the study are organized in the following. Section 2 briefs the current IDS models available for IoT networks and the design principles involved in DL based IDS is given in Section 3. Followed by, the proposed model is derived in Section 4 and the stimulation results are discussed in Section 5. At last, the conclusions are drawn in Section 6.

2 Literature Review

Ma et al. [11] presented a novel hybrid technique comprising deep neural network (DNN) and spectral clustering (SC), called spectral clustering deep neural network (SCDNN) for detecting network intrusions. The SCDNN method comprises deep neural network (DNN) and spectral clustering (SC). Initially, SC method separates the input trained datasets to k-trained subset, and later the trained subset is utilized for training the k sub-DNN classification. Then, the tested dataset composes the subset with SC and is utilized for testing the equivalent sub-DNNs. The investigational outcome demonstrated that SCDNN method has higher detecting accuracy compared to RF (random forest), BPNN (backpropagation neural network), Bayesian, and SVM techniques. Lopez-Martin et al. [12] developed an unsupervised network intrusion detection technique depending upon conditional VAE, so-called ID-CVAE. These techniques have a certain framework that combines intrusion tags within the decoder layer. The investigational outcome illustrates that the presented ID-CVAE method gives enhanced classifier outcomes compared to familiar classification. Especially, this technique can retrieve the lacking features from inadequately trained datasets. A DL technique [13] for intrusion detection utilizing the Recurrent Neural Network (RNN), known as RNN-IDS.

Li et al. [14] introduced a DL method for automatically learn the features of graphic NSL-KDD conversion by the presented graphic transformation method. They utilized Google Net, ResNet, and Convolutional Neural Network (CNN) to detect intrusion networks. The efficiency of the image transformation technique is calculated *via* binary classification testing on the NSL-KDD dataset. A new DL classification method is known as S-NDAE is presented in [15]. The presented S-NDAE utilizes RF and stacked non-symmetric deep auto-encoders (NDAEs) for building a classification method to detect intrusion. The NDAE method is utilized for learning features and thus, the stacked RF and NDAE methods are utilized for performing classification. A novel two-stage DL method depending upon stacked auto-encoders with softmax classification for detecting intrusions in [16]. This method involves 2 decision making stages: Initially, the network traffic is categorized by utilizing normal/abnormal probability values. Later, the probability values are added with the input features as an extra features for detecting normal and another kinds of attacks. For evaluating the efficiency of the presented method, wide-ranging researches have been executed on UNSW-NB15 and KDD99 datasets.

In Li et al. [17], researchers utilized the GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory) with the BLS (Extended Learning System) and variable number of hidden layers and its expansion for building the network abnormality identification method. The NSL-KDD and BGP datasets are utilized for evaluating the efficiency of the presented method. On NSL-KDD dataset, the investigational outcome illustrates that the higher efficiency is attained by utilizing GRU3 RNN and

LSTM4, and the CFBLS (BLS with cascade of mapping features) framework can give the optimum outcomes. Vinayakumar et al. [18] proposed the distributed DNN method for developing hybrid and scalable intrusion detection method is known as scale-hybrid-IDS-AlertNet (SHIA). The presented SHIA can efficiently observe the huge amount of host level and network level performance for automatically recognize malicious attacks to give network administrators proper signals. The severe investigational testing on several benchmark IDS dataset shows that the presented method can execute well related to other conventional ML classification.

3 Design Principles Involved in DL Based IDS for IoT Networks

The goal of DL enabled IDS for IoT networks is to create a model which performs well with respect to efficacy and proficiency. But every individual model accepts certain design choices which may restrict the capability in attaining the aim. For instance, few of the DL models don't address the overfitting issue or use the model on the unbalanced datasets, or disregard feature engineering processes that destructively affect the outcome with respect to accuracy, complexity, and resource utilization [19]. Besides, few of the IDS do not optimize the learning technique and few of them are validated on irrelated dataset, which does not reflect real-time IoT networking data. With this motivation, the DL enabled IDS solution for IoT needs to follow the design principles listed below.

(i) Hande overfitting: overfitting occurs once the model realizes an upright fit on the training dataset, however, it could not exhibit generalization on unknown data. The DL models can avoid overfitting by the use of regularization (adding a cost to the loss function) and dropout layers (arbitrarily discard particular features by fixing it to 0).

(ii) Balancing dataset: Imbalanced data denotes an inequality distribution of class instances in the dataset. When a model undergoes training on the imbalanced dataset, it becomes biased and favors the majority class labels. Through the proper balance of the datasets, the efficacy of the model can be enhanced.

(iii) Feature engineering: Enables cost minimization of the DL model with respect to time and memory utilization. It additionally enhances the accurateness of the model by the elimination of irrelated feature and applies feature transformation for increasing the performance of the learning technique.

(iv) Model optimization: It aims at the minimization of loss function that determines the variation among the predicted and original outcomes. It can be obtained through the iterative adjustment of the model weights. The utilization of optimization algorithms results in an increase in model efficacy.

(v) Validation: The DL enabled IDS for IoT needs to be validated on the IoT dataset for getting the results reflecting the real-time IoT networking data.

4 The Proposed IMFSDL-IDS Model

The overall system framework of the presented IMFSDL-IDS model is demonstrated in Fig. 1. The figure reveals that IoT devices are initially employed to acquire network data from the target environment. For handling big data, Hadoop ecosystem is employed. Then, the preprocessing of data take place to transform the data into a compatible format. Next to that, the HCMSO based feature subset selection process is performed to select an optimum subset of features. Lastly, the classification method is executed by the use of BAS-VAE model, and thereby detect the existence of intrusions in the IoT networks. The detailed working of every component in the system architecture is discussed in the subsequent sections.

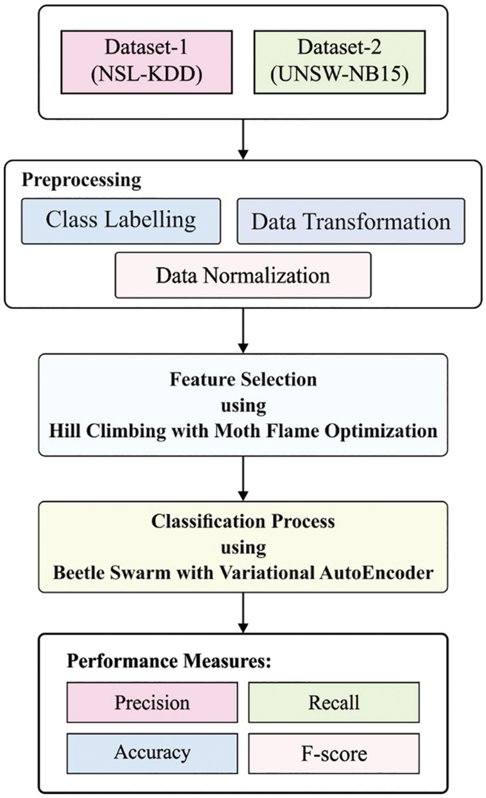

Figure 1: Overall working process of IMFSDL-IDS model

4.1 Hadoop Ecosystem

For the management of Big Data, Hadoop Ecosystem and the respective elements are widely applied. On the IoT networks, Hadoop allows the user to store and analyze the big data over clusters through simple programming models. It offers scalable and faults tolerant towards by the use of 1000's nodes from an individual server [20]. A set of three major components involved in Hadoop are MapReduce, Hadoop Distributed File System (HDFS), and Hadoop YARN.

4.2 Data Preprocessing

The IMFSDL-IDS model receives only numerical values to train and test the model. Therefore, 1-hot encoding method is applied for transferring every symbolic feature value on the dataset into numerical ones [21]. Besides, every symbolic attribute is transformed into numerical values. Then, data normalization is applied for the normalization of the range of data feature that can fasten the algorithm execution. Here, maximum-minimum normalization technique is applied for scaling the feature values. Every feature value undergoes normalization to a particular interval of [0,1] using Eq. (1).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x and x' refers to actual and normalized values respectively.

4.3 Feature Subset Selection

The preprocessed data is fed into the HCMFO algorithm to elect proper subset of features. Moths have been developed for flying in the night-time by utilizing the moonlight and they are based on this approach is known as transverse orientation for navigation. In this technique stimulated from this kind of movement is introduced. Moths and flames are the major elements of this method [22]. The candidate solution is moths and the moth's locations in space are the problems parameters. Thus, moths could fly in one dimension, two dimensions, three dimensions, or even in hyper dimension space (i.e., dimension d) by varying location vectors. Flames are the optimum n location of moth that is previously attained. Consequently, flames represent d-dimension data point. Considering logarithmic spiral, the moth upgrades its location regarding the provided flame in Eq. (2).

$$S\left(M_i,\ F_j\right) = D_i \cdot e^{bt}.\cos(2\pi t) + F_j \tag{2}$$

where D_i implies Euclidian distance of the i^{th} moth for j^{th} flame, b represents constant for determining the shape of the logarithmic spiral, M_i indicates i^{th} moth, F_j denotes j^{th} flame and t represent arbitrary numbers in [−1, 1].

In spiral formula, t variable determines the following location of the moth, which must be adjacent to the flame. Thus, a hyper ellipse is considered near the flame in every direction and the following location of moth should be inside the space. Additional emphasize the exploitation, it considers t as a random number in $[r,\ 1]$ in which r denotes linear reduction from −1 to −2 across the iteration process, is known as convergence constant. By this technique, moths consider the exploitation which is equivalent to the flames are more precisely relative to the iteration counts. Additionally, for enhancing the possibility of converging to a global solution, an assumed moth is required for updating the location by utilizing individual flames. In all iterations and then upgrading the flame list, the flames are arranged to depend upon fitness values. The moths after updating their locations in relation to their equivalent flames. For allowing too much exploitation of the optimum solution, the flame counts should be reduced regarding the iteration amount as in Eq. (3). MFO is given in the Eq. (2).

$$N_{flames} = round\left(N - l \cdot \frac{N-1}{T}\right) \tag{3}$$

where l represents present iteration number, N indicates maximum flame counts, and T denotes maximum iteration counts.

The presented MFO for FS process in a wrapper based method. The key feature of wrapper method is the classification usage as a guideline for FS process. In this presented technique, it utilized VAE as a classification technique for ensuring the quality of the selected feature sets. VAE is a simpler classification model which categorizes the unknown sample. This research utilizes the presented MFO technique to adaptively search the feature space for optimum feature composite, maximize the classification efficiency. In MFO method, the moth always changing their locations at all points in the space based on the spiral Eq. (2). The t variable is chosen at arbitrary and decide the following location of moth M_i. For distinct values of t and considering the distance among flame F_j, and moth M_i the novel location of M_i could be viewed as an alteration of the location of F_j and vector δ that is represented in Eq. (4).

$$\delta^t_{M_i,F_j,d} = D_i \cdot e^{bt} \cdot cos(2\pi t) \tag{4}$$

A separate solution denotes the continuous vector with similar dimension as feature counts in the dataset. A generic illustration of the fitness function (FF) demonstrating both classification efficiency and amount of selected feature in Eq. (5).

$$f_\theta = \alpha \cdot E + (1—\alpha)\frac{\sum_i \theta_i}{N} \tag{5}$$

where f_θ denotes FF considering a vector θ sized N with zero or one component demonstrating selected or unselected features, N denotes entire feature counts in the dataset, E represents classification error rate and α indicates constant monitoring the significance of classification efficiency for the amount of selected feature. In this research, the classification efficiency is the main objective which is utilized as $\alpha = 1$. Fig. 2 illustrates the flowchart of MFO technique [23]. For enhancing the convergence efficiency of the MFO algorithm, HC concept is incorporated into it. It enables exchange of a moth earlier to the optimal location using a local minimum of the current location. The local minima have been offered by the HC. It interchanges the labels of two vertices when it recognizes the bandwidth is reduced. The HC technique primarily identifies the critical vertex and looks for the proper vertex for interchanging.

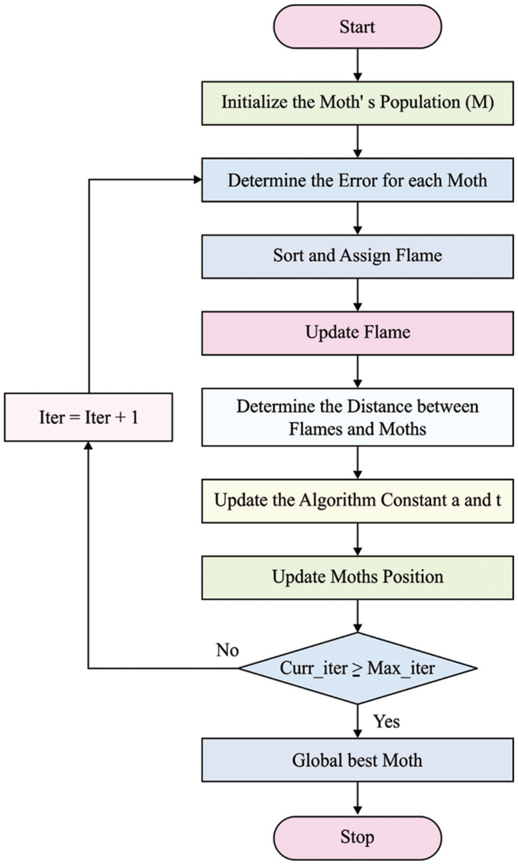
Start
Initialize the Moth' s Population (M)
Determine the Error for each Moth
Sort and Assign Flame
Update Flame
Determine the Distance between Flames and Moths
Iter = Iter + 1
Update the Algorithm Constant a and t
Update Moths Position
No
Curr_iter ≥ Max_iter
Yes
Global best Moth
Stop

Figure 2: Flowchart of MFO algorithm

4.4 Data Classification

In the last phase, the BAS-VAE technique is utilized for determining the presence of intrusion in the IoT networks.

VAE is a generative technique which provides a probability-based model to define the observation in latent space. It integrates the variational interpretation to the neural networks as function which determines the approximate posterior distribution [24]. VAEs can generate new data when the model undergoes training through the sampling process. It is carried out by the creation of the hyperparametric description of the data which is chosen to have low feature dimensions. Owing to the intrinsic probability based characteristics of the VAE model, it finds useful to design the IDS. Fig. 3 demonstrates the architecture of AE.

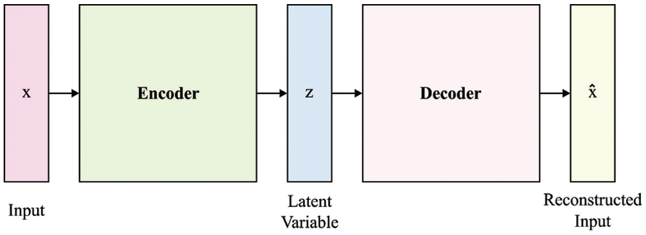

Figure 3: Structure of AutoEncoder

For a provided the dataset X comprising of N instances, signifying as $X = \{\mathrm{x}_n\}_{n=1}^{N}$, VAE determines the generative model in the latent variable z_n to x_n using the decoder with parameter θ like $p_\theta(x_n|z_n)$. Generally, the decoding is DNN and outputs the distribution parameter of $p_\theta(x_n|z_n)$. The evaluation of θ is done by maximize the lower bound $\sum_{n=1}^{N} \mathrm{LB}(\theta,\ \varphi;\mathrm{x}_n)$ of log marginal probability $\ln p_\theta(\mathrm{x}_1, \ldots,\ \mathrm{x}_N)$. The lower bound $LB(\theta,\ \varphi;\mathrm{x}_n)$ is represented as:

$$LB(\theta,\ \varphi;\mathrm{x}_n) = \mathbb{E}_{q_\varphi(z_n|x_n)}[\ln p_\theta(x_n|z_n)] - KL\big(q_\varphi(z_n|x_n)\|p_\theta(z_n)\big) \tag{6}$$

whereas the initial term of right-hand side is the assumption of log probability $lnp_\theta(x_n|z_n)$ interms of the variational distribution $q_\varphi(z_n|x_n)$, $p_\theta(z_n)$ is the preceding distribution of latent variable z_n, and $\mathrm{KL}(\cdot\|\cdot)$ refers the Kullback–Leibler (KL) divergence. In order to manage the issue of non-analytical solution of lower bound, affected by $\mathbb{E}_{q_\varphi(z_n|x_n)}[\ln p_\theta(x_n|z_n)]$, the VAE employs an encoder with parameter φ for representing $q_\varphi(z_n|x_n)$. An encoder takes x_n as the input as well as outputs the distribution parameter of z_n condition on x_n. So, the Monte Carlo estimator of lower bound in (2) is illustrated as:

$$LB(\theta,\ \varphi;\mathrm{x}_n) \simeq \frac{1}{I}\sum\nolimits_{i=1}^{I} \ln p_\theta\big(\mathrm{x}_n|\mathrm{z}_n^i\big) - KL\big(q_\varphi(z_n|x_n)\|p_\theta(z_n)\big) \tag{7}$$

where z_n^i implies the ith sampling value of z_n sampled in $q_\varphi(z_n|x_n)$. By this technique, the lower bound has a clear appearance and is optimization by acting stochastic gradient ascent (SGA). Fig. 4 showcases the structure of VAE.

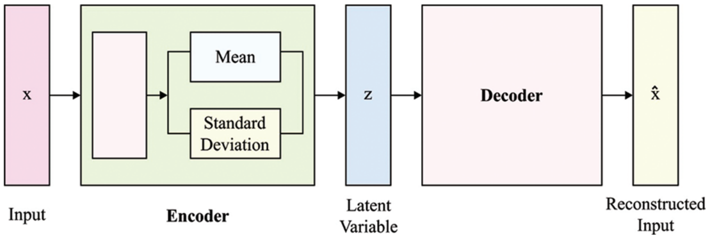

Figure 4: Structure of VAE

But, the sampling of latent variable z_n hinder the error BP that creates the execution of SGA impossible [25]. In order to manage this issue, the re-parameterization trick is taken that establishes an auxiliary parameter ε_n with the prior distribution $p(\varepsilon_n)$ and signifies z_n as

$$z_n = g_\varphi(x_n,\ \varepsilon_n), \varepsilon_n \sim p(\varepsilon_n) \tag{8}$$

For instance, getting Gaussian distribution as posterior distribution of z_n by their mean $\mu_\varphi(x_n)$ and standard deviation (SD) $\sigma_\varphi(x_n)$ even encoded by encoder, afterward, it contains $z_n = \mu_\varphi(x_n) + \sigma_\varphi(x_n) \odot \varepsilon_n$ with $p(\varepsilon_n) = \mathcal{N}(\varepsilon_n|0,\ I)$, $\mathcal{N}(\cdot)$ referring the Gaussian distribution, and $\odot$ become the elementwise product operators. The sampling of z_n is written as $z_n^i = \mu_\varphi(x_n) + \sigma_\varphi(x_n) \odot \varepsilon_n^i$, where ε_n^i implies the ith sampling value of ε_n. The parameterization model prevents the direct sampling of z_n, so permitting BP with the latent variable. To further improve the detection efficiency of the VAE, the model parameter optimization process is carried out using BAS algorithm. The BAS technique utilizes 2 principles simulated from the performance of beetle searching process with antennae. It can be noticeable the beetle searches arbitrarily for exploring an unknown location [26]. The steps involved in BAS are given as follows.

Step 1: Let the location of longicorn beetles in n-dimensional solution space is $= (x_1,\ x_2, \ldots, x_n)$, to design the searching performance and the arbitrary process of beetle search can be defined as:

$$\vec{p} = \frac{\text{rand}(n, 1)}{\|\text{rand}(n, 1)\|} \tag{9}$$

where rand $(n,\ 1)$ implies the n-dimensional vector of random number within zero and one.

Step 2: Present the search performance together of right-hand as well as left-hand side correspondingly, for imitating the actions of beetle's antennae:

$$\begin{cases} x_l^k = x^k - \mathrm{d}\vec{p}, \\ x_r^k = x^k + \mathrm{d}\vec{p}, \end{cases} \tag{10}$$

where x^k refers the present location of longicorn beetles, d implies the distance in the center of mass to the antennae, x_r^k indicates a location lying from the searching region of right-hand side, and x_l^k represents the left-hand side.

Step 3: Location upgrade model:

$$x^{k+1} = x^k + \vec{p}\ \delta^k sign\left(f\left(x_l^k\right) - f\left(x_r^k\right)\right) \tag{11}$$

where δ^k represents the current step size, *sign* implies the symbolic function, and f refers the function as optimization.

5 Performance Validation

This section offers the experimental validation of the IMFSDL-IDS technique on two standard datasets namely NSL-KDD and UNSW-NB15 dataset. The former dataset includes a total of 3333 samples, 41 attributes, and 2 classes. Likewise, the latter dataset comprises a set of 257673 samples, 49 attributes, and 2 classes. For experimentation, tenfold crossvalidation technique is employed. The details associated with the datasets are shown in Tab. 1.

Tab. 2 illustrates the FS results analysis of the HCMFO algorithm on the applied two datasets. The table values demonstrated that the HCMFO algorithm has chosen a subset of features from the NSL-KDD and UNSW-NB15 dataset with the least good cost of 0.002468 and 0.003467 respectively.

Table 1: Dataset description

Description	NSL-KDD	UNSW-NB15
Number of Instances	3333	257673
Number of Features	41	49
Number of Class	2	2
Percentage of Normal Samples	67343	93000
Percentage of Attack Samples	58630	164673

Table 2: Results analysis of HCMFO based feature selection on applied dataset

Dataset	Best cost	Selected features
NSL-KDD	0.002468	1, 2, 3, 6, 8, 21, 23, 26, 28, 31, 33, 36, 37, 39, 40
UNSW-NB15	0.003467	1, 3, 5, 7, 13, 17, 19, 20, 21, 25, 27, 29, 30, 32, 33, 36, 38, 39, 40, 42, 43, 44, 45, 48

Tab. 3 and Fig. 5 inspect the intrusion detection performance of the IMFSDL-IDS model on the employed NSL-KDD dataset. From the obtained values, it is evident that the IMFSDL-IDS technique has accomplished improved intrusion detection performance under different types of attacks. For instance, the IMFSDL-IDS model has detected the DoS attacks with the prec. of 87.46%, rec. of 95.10%, F1-score of 91.32%, and acc. of 96.23%. Eventually, the IMFSDL-IDS method has detected the R2l attacks with the prec. of 87.23%, rec. of 96.80%, F1-score of 90.51%, and acc. of 95.26%.

Table 3: Result analysis of proposed IMFSDL-IDS method on NSL-KDD dataset

Attack type	Precision	Recall	F1-score	Accuracy
Dos	87.46	95.10	91.32	96.23
R2l	87.23	96.80	90.51	95.26
Probe	86.31	95.62	90.73	95.01
U2r	87.53	95.21	89.98	94.91
Normal	87.28	94.26	90.25	94.86
Average	87.16	95.40	90.56	95.25

Meanwhile, the IMFSDL-IDS approach has detected the Probe attacks with the prec. of 86.31%, rec. of 95.62%, F1-score of 90.73%, and acc. of 95.01%. Concurrently, the IMFSDL-IDS method has detected the U2r attacks with the prec. of 87.53%, rec. of 95.21%, F1-score of 89.98%, and acc. of 94.91%. Simultaneously, the IMFSDL-IDS methodology has detected the Normal attacks with the prec. of 87.286%, rec. of 94.26%, F1-score of 90.25%, and acc. of 94.86%.

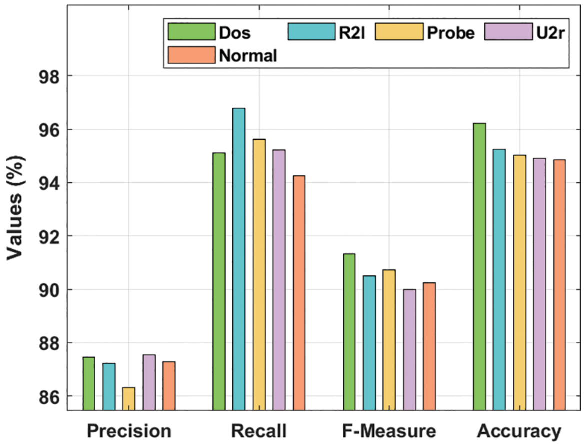

Figure 5: Result analysis of IMFSDL-IDS model on NSL-KDD dataset

Tab. 4 and Fig. 6 examine the intrusion detection performance of the IMFSDL-IDS method on the applied UNSW-NB15 dataset. From the achieved values, it is apparent that the IMFSDL-IDS approach has accomplished enhanced intrusion detection performance under distinct types of attacks. For instance, the IMFSDL-IDS method has detected the Normal attacks with the prec. of 88.82%, rec. of 97.18%, F1-score of 92.24%, and acc. of 97.36%.

Table 4: Result analysis of proposed IMFSDL-IDS method on UNSW-NB15 dataset

Attack type	Precision	Recall	F1-score	Accuracy
Normal	88.82	97.18	92.24	97.36
DoS	89.42	96.44	91.83	97.85
Shellcode	90.12	97.25	91.87	95.68
Backdoor	88.46	96.57	91.99	97.89
Generic	89.34	96.72	91.04	98.57
Exploits	90.25	95.79	92.31	95.91
Fuzzers	89.16	95.49	91.92	98.65
Reconnaissance	88.89	95.31	89.61	97.99
Analysis	88.17	96.79	91.87	97.99
Worms	89.02	96.42	90.64	96.05
Average	89.17	96.40	91.53	97.39

Also, the IMFSDL-IDS technique has detected the DoS attacks with the prec. of 89.42%, rec. of 96.44%, F1-score of 91.83%, and acc. of 97.85%. Besides, the IMFSDL-IDS approach has detected the Shellcode attacks with the prec. of 90.12%, rec. of 97.25%, F1-score of 91.87%, and acc. of 95.68%. Followed by, the IMFSDL-IDS manner has detected the Backdoor attacks with the prec. of 88.46%, rec. of 96.57%, F1-score of 91.99%, and acc. of 97.89%. Additionally, the IMFSDL-IDS approach has detected the Generic attacks with the prec. of 89.34%, rec. of 96.72%, F1-score of 91.04%, and acc. of 98.57%. Moreover, the IMFSDL-IDS methodology has detected the Exploits attacks with the prec. of 90.25%, rec. of 95.79%, F1-score of 92.31%, and acc. of 95.91%. Furthermore, the IMFSDL-IDS technique has detected the Fuzzers attacks with the prec. of 89.16%, rec. of 95.49%, F1-score of 91.92%, and acc. of 98.65%. Subsequently, the IMFSDL-IDS technique has detected the Reconnaissance attacks with the prec. of 88.89%, rec. of 95.31%, F1-score of 89.61%, and acc. of 97.99%. Along with that, the IMFSDL-IDS algorithm has detected the Analysis attacks with the prec. of 88.17%, rec. of 96.79%, F1-score of 91.87%, and acc. of 97.99%. At last, the IMFSDL-IDS technique has detected the Worms attacks with the prec. of 89.02%, rec. of 96.42%, F1-score of 90.64%, and acc. of 96.05%.

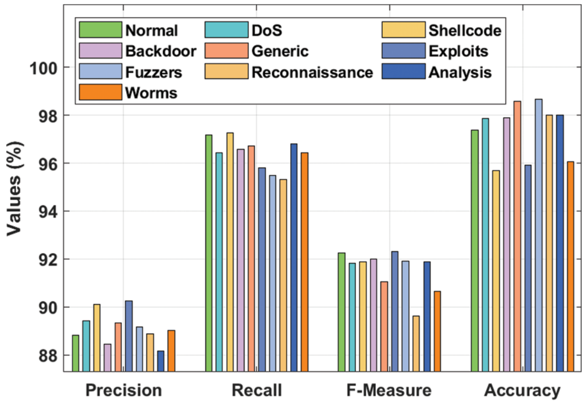

Figure 6: Result analysis of IMFSDL-IDS model on UNSW-NB15 dataset

Tab. 5 and Fig. 7 offer the comparative results analysis of the IMFSDL-IDS model with the existing methods on the applied NSL-KDD dataset. From the table, it is observed that the EM model has exhibited worse performance over the other methods. At the same time, the Standard MLP-SM1, TSDL -ANN, GALR-DT, and RF methods have demonstrated slightly increased results over the EM model. Followed by, the standard MLP-NB, TSDL-LR, NAWIR, TSDL-DT, and standard MLP-SM2 models have portrayed closer and moderate performance.

Table 5: Comparative analysis of proposed IMFSDL-IDS method on NSL-KDD dataset

Model	Precision	Recall	F1-score	Accuracy
IMFSDL-IDS	87.16	95.40	90.56	95.25
BAS-VAE	86.96	95.03	90.12	94.95
FFN	86.76	94.24	89.75	94.70
VAE	87.90	95.21	90.20	93.30
Two-stage Ensemble	91.60	91.30	–	91.72
GALR-DT	–	–	–	81.42
NAWIR	–	98.50	–	83.47
RF	–	81.60	79.50	81.61
Standard MLP-SM1	–	–	–	81.30
Standard MLP-SM2	–	63.27	–	89.13
Standard MLP-NB	–	–	–	82.07
TSDL -DT	–	–	–	85.56
TSDL -ANN	–	–	–	81.34
TSDL -LR	–	–	–	83.15
EM	–	–	–	78.47

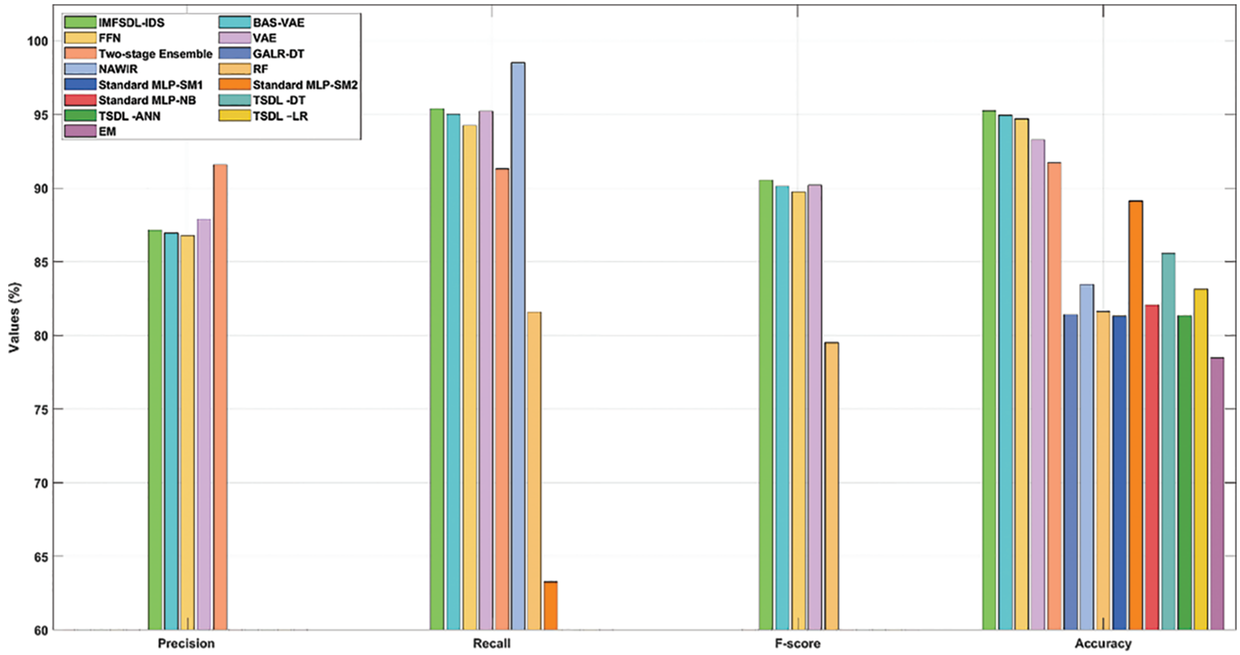

Figure 7: Comparative analysis of IMFSDL-IDS model on NSL-KDD dataset

Along with that, the two-stage ensemble and VAE models have depicted manageable outcomes whereas even increased results are accomplished by the FFN and BAS-VAE model. However, the proposed IMFSDL-IDS model has demonstrated effective performance with the maximum precision of 87.16%, recall of 95.40%, F1-score of 90.56%, and accuracy of 95.25%.

Tab. 6 and Fig. 8 give the comparative outcomes analysis of the IMFSDL-IDS method with the existing methods on the applied UNSW-NB15 dataset. From the table, it can be stated that the STL-IDS method has demonstrated least performance over the other techniques. Likewise, the DBN, S-NDAE, SVM, and Two-stage Ensemble techniques have portrayed somewhat higher outcomes over the STL-IDS model. Besides, the standard Ensemble, Multilayer, and DBN+SVM approaches have showcased closer and moderate performance. In addition, the two-stage ensemble and VAE models have depicted manageable results whereas even increased results are accomplished by the FFN and BAS-VAE models.

Table 6: Comparative Analysis of Proposed IMFSDL-IDS Method on UNSW-NB15 Dataset

Model	Precision	Recall	F1-score	Accuracy
IMFSDL-IDS	89.17	96.40	91.53	97.39
BAS-VAE	89.06	96.10	91.02	96.98
FFN	88.20	95.86	90.96	96.70
VAE	87.90	95.42	91.30	97.01
Two-stage Ensemble	88.00	86.80	–	85.79
DBN	–	80.58	84.08	80.58
S-NDAE	–	85.42	87.37	85.42
SVM	–	86.22	89.30	–

(Continued)

Table 6 (continued)

Model	Precision	Recall	F1-score	Accuracy
Ensemble	–	90.45	93.91	–
Multilayer	–	91.98	94.36	–
DBN + SVM	–	92.17	94.65	–
STL-IDS	–	76.56	79.07	80.48

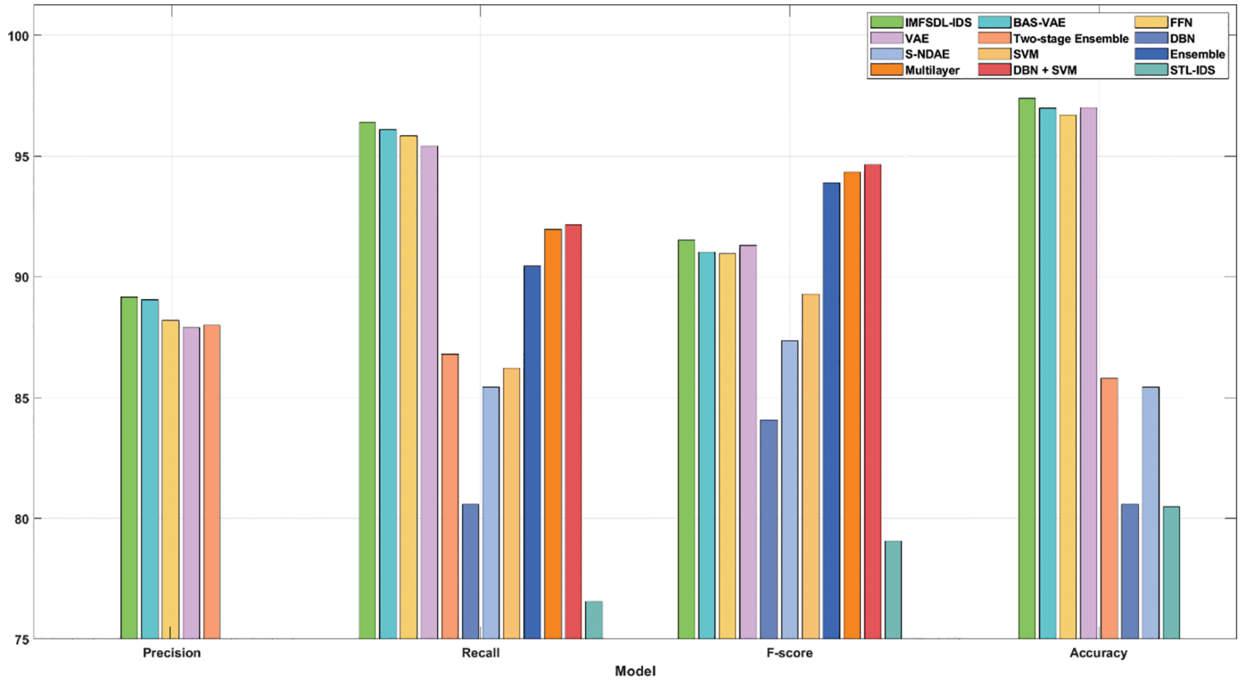

Figure 8: Comparative analysis of IMFSDL-IDS model on UNSW-NB15 dataset

However, the proposed IMFSDL-IDS model has outperformed effective performance with a higher precision of 89.17%, F1-score of 91.53%, recall of 96.40%, and accuracy of 97.39%. For assessing the effectual intrusion detection results of the IMFSDL-IDS technique, a series of experimentation takes place and the results are inspected with respect to several dimensions. The resultant experimental values pointed out the betterment of the IMFSDL-IDS models over the compared models interms of disinct measures.

6 Conclusion

This paper has presented an innovative IMFSDL-IDS method to effectively detect intrusions in the IoT networks. The proposed IMFSDL-IDS model primarily utilizes IoT devices for data acquisition. Followed by, data preprocessing is carried out in two levels such as data transformation and data normalization. Next, the feature subset selection procedure is performed using HCMFO algorithm. Finally, the BAS-VAE technique is employed for the detection of intrusions in IoT network. The BAS algorithm is integrated into the VAE to properly tune the parameters involved in it and thereby raises the classification performance. For assessing the effectual intrusion detection results of the IMFSDL-IDS model, a series of experimentation takes place and the outcomes are inspected with respect to several dimensions. The resultant experimental values pointed out the betterment of the IMFSDL-IDS model over the compared methods with the maximal 95.25% and 97.39% accuracy on the applied UNSW-NB15 and NSL-KDD dataset respectively. As a part of future extension, outlier detection techniques can be incorporated into the presented model to eradicate the existence of outliers in the real-time networking data.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] 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.

[2] A. R. Javed, S. u. Rehman, M. U. Khan, M. Alazab and T. R. G, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1456–1466, 2021.

[3] H. Tahaei, F. Afifi, A. Asemi, F. Zaki and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *Journal of Network and Computer Applications*, vol. 154, pp. 102538, 2020.

[4] S. Hajiheidari, K. Wakil, M. Badri and N. J. Navimipour, "Intrusion detection systems in the Internet of things: A comprehensive investigation," *Computer Networks*, vol. 160, no. 5, pp. 165–191, 2019.

[5] S. Neelakandan, R. Annamalai and M. D. Kumar, "Efficient solution to the waste management process using iot for smart trash can," *Journal of Emerging Technologies and Innovative Research*, vol. 5, no. 6, pp. 426–427, 2018.

[6] E. Spafford, "James, p. anderson: An information security pioneer," *IEEE Security and Privacy*, vol. 6, no. 1, pp. 1–9, 2008.

[7] M. S. Alnaghes and F. Gebali, "A survey on some currently existing intrusion detection systems for mobile ad hoc networks," in *Proc. of the Second Int. Conf. on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC2015)*, Antalya, Turkey, vol. 12, pp. 26–28, 2015.

[8] J. Uthayakumar, N. Metawa, K. Shankar and S. K. Lakshmanaprabu, "Intelligent hybrid model for financial crisis prediction using machine learning techniques," *Information Systems and e-Business Management*, vol. 18, no. 4, pp. 617–645, 2020.

[9] S. Neelakandan and D. Paulraj, "An automated learning model of conventional neural network based sentiment analysis on twitter data," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 5, pp. 2230–2236, 2020.

[10] J. Uthayakumar, D. Nivetha, D. Vinotha and M. Vasanthi, "Classification rule discovery using ant-miner algorithm: An application of network intrusion detection," *International Journal of Modern Engineering Research*, vol. 4, no. 8, pp. 70–83, 2014.

[11] T. Ma, F. Wang, J. Cheng, Y. Yu and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, pp. 1701, 2016.

[12] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT," *Sensors*, vol. 17, no. 9, pp. 1967, 2017.

[13] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[14] Z. Li, Z. Qin, K. Huang, X. Yang and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Int. Conf. on Neural Information Processing, ICONIP 2017*, Cham, Switzerland, Springer, pp. 858–866, 2017.

[15] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[16] F. A. Khan, A. Gumaei, A. Derhab and A. Hussain, "TSDL: A twostage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[17] Z. Li, A. L. G. Rios, G. Xu and L. Trajković, "Machine learning techniques for classifying network anomalies and intrusions," in *2019 IEEE Int. Sym. on Circuits and Systems (ISCAS)*, Sapporo, Japan, pp. 1–5, 2019.

[18] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. A. Nemrat *et al.,* "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[19] A. Derhab, A. Aldweesh, A. Z. Emam and F. A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, pp. 1–16, 2020.

[20] R. T. Selvi and I. Muthulakshmi, "Modelling the map reduce based optimal gradient boosted tree classification algorithm for diabetes mellitus diagnosis system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1717–1730, 2021.

[21] Y. Yang, K. Zheng, B. Wu, Y. Yang and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020.

[22] H. M. Zawbaa, E. Emary, B. Parv and M. Sharawi, "Feature selection approach based on moth-flame optimization algorithm," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, pp. 4612–4617, 2016.

[23] M. Talaat, A. S. Alsayyari, M. A. Farahat and T. Said, "Moth-flame algorithm for accurate simulation of a non-uniform electric field in the presence of dielectric barrier," *IEEE Access*, vol. 7, pp. 3836–3847, 2019.

[24] M. A. Albahar and M. Binsawad, "Deep autoencoders and feedforward networks based on a new regularization for anomaly detection," *Security and Communication Networks*, vol. 2020, no. 8, pp. 1–9, 2020.

[25] J. Chen, L. Du and L. Liao, "Discriminative mixture variational autoencoder for semisupervised classification," *IEEE Transactions on Cybernetics*, pp. 1–15, 2020. Article in press, https://doi.org/10.1109/TCYB.2020.3023019.

[26] H. He, S. Zhou, L. Zhang, J. Lin, W. Chen *et al.,* "Beetle swarm optimization algorithm-based load control with electricity storage," *Journal of Control Science and Engineering*, vol. 2020, no. 17, pp. 1–8, 2020.