Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.036352
*Article*

# Optimal Deep Learning Based Intruder Identification in Industrial Internet of Things Environment

**Khaled M. Alalayah[1], Fatma S. Alrayes[2], Jaber S. Alzahrani[3], Khadija M. Alaidarous[1], Ibrahim M. Alwayle[1], Heba Mohsen[4], Ibrahim Abdulrab Ahmed[5] and Mesfer Al Duhayyim[6,*]**

[1]Department of Computer Science, College of Science and Arts, Sharurah, Najran University, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Saudi Arabia
[4]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[5]Computer Department, Applied College, Najran University, Najran, 66462, Saudi Arabia
[6]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, 16273, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa

Received: 27 September 2022; Accepted: 25 November 2022

**Abstract:** With the increased advancements of smart industries, cybersecurity has become a vital growth factor in the success of industrial transformation. The Industrial Internet of Things (IIoT) or Industry 4.0 has revolutionized the concepts of manufacturing and production altogether. In industry 4.0, powerful Intrusion Detection Systems (IDS) play a significant role in ensuring network security. Though various intrusion detection techniques have been developed so far, it is challenging to protect the intricate data of networks. This is because conventional Machine Learning (ML) approaches are inadequate and insufficient to address the demands of dynamic IIoT networks. Further, the existing Deep Learning (DL) can be employed to identify anonymous intrusions. Therefore, the current study proposes a Hunger Games Search Optimization with Deep Learning-Driven Intrusion Detection (HGSODL-ID) model for the IIoT environment. The presented HGSODL-ID model exploits the linear normalization approach to transform the input data into a useful format. The HGSO algorithm is employed for Feature Selection (HGSO-FS) to reduce the curse of dimensionality. Moreover, Sparrow Search Optimization (SSO) is utilized with a Graph Convolutional Network (GCN) to classify and identify intrusions in the network. Finally, the SSO technique is exploited to fine-tune the hyper-parameters involved in the GCN model. The proposed HGSODL-ID model was experimentally validated using a benchmark dataset, and the results confirmed the superiority of the proposed HGSODL-ID method over recent approaches.

**Keywords:** Industrial IoT; deep learning; network security; intrusion detection system; attribute selection; smart factory

## 1 Introduction

With the gradual advancements of informatization and industrialization, the safety and controllability of the industrial Internet of things (IIoT) have gained significant interest among research communities. The main idea behind the development of IIoT is to reap the advantages of Internet of Things (IoT) technologies and apply it in Industrial Control Systems (ICSs). ICSs have become an essential part of critical structures. ICSs are used for a known period in monitoring industrial machinery and the associated processes [1]. It accomplishes real-time observation and communication with machines, performs real-time data collection and analysis, and keeps a log of every activity of the industrial systems. The application of IoT in such mechanisms improves the network's security and brings intelligence to the automation and optimization of industrial progressions [2]. Supervisory Control and Data Acquisition (SCADA) mechanism is one of the major components of ICSs. It offers a Graphical User Interface (GUI) via Human Machine Interface (HMI) [3]. HMI eases the processes, for an operator, in terms of system status monitoring, communication with IIoT gadgets, and triggering the alarm in case of abnormal actions. Network intrusion is any effort to destroy the integrity, availability, or confidentiality of the network and its host [4]. It is considered the most typical menace in cyberspace. This is because the existing intrusion prevention conditions are insufficient and static in nature. Conventional Network Intrusion Detection (ID) techniques are generally passive and cannot efficiently identify different types of unknown intrusions [5,6]. Thus, developing a precise and effective intellectual network ID approach is imperative.

An intrusion Detection System (IDS) is a network security gadget that observes real-time network trafficking and triggers a warning or takes proactive actions in case of any suspicious communications [7]. IDSs vary from other prevention mechanisms by detecting the ongoing invasion or invasion that happened earlier [8]. ID can generally be modelled as a binary classification issue that differentiates whether a network traffic conduct is anomalous or normal or a multi-class classifier issue, in which the network traffic conduct is recognized and the network attack type is fixed [9]. Over the past few years, various developments have occurred in the field of Artificial Intelligence (AI), like Deep Learning (DL) and Machine Learning (ML) methods that aim to enhance IoT IDS. The existing demands have been discussed up-to-date via a critical review and the taxonomy of literature [10,11]. Several relevant researchers have applied different ML and DL methods, using numerous datasets, to execute and authenticate the enhancement of IoT IDS. However, it is still a debate whether the ML method or DL method is highly efficient in framing a potential IoT IDS [12]. In literature, the time taken to develop, train, test, and validate an IoT IDS was not considered so far to assess certain IDSs approaches. However, this is an important factor that decides the efficiency of online IDSs [13]. Recently, DL methods have gained popularity since they can resolve network ID issues [14].

In this background, the current study introduces a Hunger Games Search Optimization with Deep Learning-Driven Intrusion Detection (HGSODL-ID) model in an IIoT environment. The presented HGSODL-ID model exploits the linear normalization approach to transform the input data into a useful format. The HGSO algorithm is employed for Feature Selection (HGSO-FS) to reduce the curse of dimensionality. Moreover, Sparrow Search Optimization (SSO) is utilized with Graph Convolutional Network (GCN) for classification and identifying intrusions in the network. Finally, the SSO method is exploited to fine-tune the hyper-parameters related to the GCN model. The proposed HGSODL-ID approach was experimentally validated using a benchmark dataset. In short, the paper's contribution is summarized as follows.

- Develop a new HGSODL-ID technique for intrusion detection in the IIoT environment.
- Design a new HGSO-FS technique for the feature selection process.

- Employ GCN-based classification with an SSO-based hyperparameter tuning process to improve the detection rate.

## 2 Literature Review

Awotunde et al. [15] presented a DL-based ID paradigm for IIoT, with hybrid rule-based Feature Selection (FS), for training and verifying the dataset. The trained method was executed using a hybrid rule-based FS and Deep Feedforward Neural Network (FFNN) technique. Li et al. [16] presented a DL technique for ID in which the Multi-Convolutional Neural Network (CNN) fusion approach was used. Based on the correlation values, the feature data was separated into four portions, after which 1D feature data was transformed into a grayscale graph. By employing flow data visualization system, CNN was established as a solution for ID problems with four optimal outcomes. In the study conducted earlier [17], the authors presented a forensic-based DL technique (termed Deep-IFS) for intrusion recognition in IIoT traffic. This method learnt local representation with the help of the Local Gated Recurrent Unit (LocalGRU) and established a Multihead Attention (MHA) layer to capture and learn about global representations (e.g., long-range dependency). A residual connection was planned amongst the layers to prevent data loss. One of the important challenges faced by the present IIoT forensics structure is its restricted scalability, which limits its efficiency in controlling huge volumes of IIoT traffic data generated by IIoT devices.

Al-Hawawreh et al. [18] introduced a detection method based on DL approaches, while the model was trained and tested using the data retrieved from the Remote Telemetry Unit (RTU) streams of the gas pipeline model. The model employed the Sparse and Denoising Autoencoder (AE) approach upon unsupervised learning. In contrast, Deep Neural Network (DNN) was employed upon supervised learning to generate high-level data representation in unlabelled and noisy data. Gyamfi et al. [19] presented a lightweight IDS based on the Online Incremental Support Vector Data Description (OI-SVDD) anomaly detection method on IIoT devices and Adaptive Sequential Extreme Learning Machine (AS-ELM) on Multi-access Edge Computing (MEC) servers. Furthermore, the authors employed MEC servers that offered computational resources to execute the AS-ELM technique at network edges. Fatani et al. [20] established an extraction feature and selective approaches for the IDS model with the help of the SI technique. The authors designed a feature extraction process based on CNNs. Afterwards, the authors examined an alternative FS technique with the help of the newly-established SI technique, i.e., Aquila Optimizer (AQU).

## 3 The Proposed Model

In the current research, an HGSODL-ID technique has been proposed for detecting and classifying intrusions in the IIoT environment. The presented HGSODL-ID model follows a series of sub-processes: linear normalization, HGSO-FS-based feature selection, GCN classification, and SSO-based hyperparameter optimization. HGSO-based feature selection and SSO-based optimization of the parameters increase the detection performance of the HGSODL-ID model. Fig. 1 depicts the block diagram of the HGSODL-ID approach.

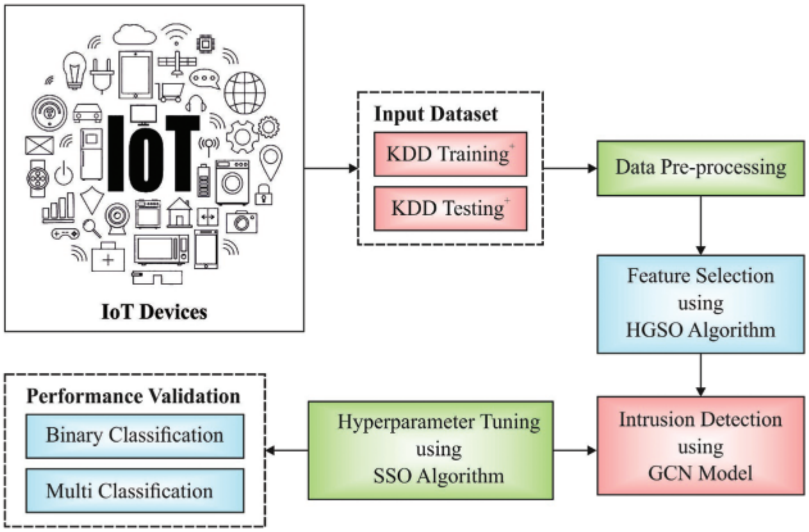

**Figure 1:** Block diagram of the HGSODL-ID approach

### *3.1 Data Pre-processing*

Initially, the presented HGSODL-ID model exploits the linear normalization approach to transform the input data into a useful format. The data should be normalized so that the dataset in the sample lies in the interval of 0 and 1. Since the dataset generally contains normal and anomalous traffic, avoiding the adverse effects of sample mean and variance is important. A simple linear normalization function is employed herewith for numerical features.

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

In Eq. (1), $\chi_{min}$ and $\chi_{max}$ signify the minimal and maximal values from each dataset, respectively. For feature 'duration', '*src_bytes*', and *dst_bytes*' are given, while the data range is large, due to which logarithmic normalization is needed.

### *3.2 Process Involved in HGSO-FS Technique*

In this study, the HGSO-FS technique is employed to reduce the curse of dimensionality. HGSO algorithm imitates the animal hunger-driven action and their behavioural preference. The model was proposed by Yang et al. recently [21]. During food search, an animal shows two social behaviours; initially, the animal cooperates in a group; in the next phase, some individuals do not participate in collaborative action. To simulate social performance, the subsequent formula is used.

$$Z_i\left(t^{\rightarrow} + 1\right) = \begin{cases} Game\,1 z^{\rightarrow}_{i(t)} \cdot \left(1 + randn\left(1\right)\right) & r_1 < L, \\ Game\,2\colon \vec{W}_{1i}\left(t\right) \cdot \vec{Z_i^b}\left(t\right) & \\ +\vec{C}_i\left(t\right) \cdot \vec{W}_{2i}\left(t\right) \cdot \left|\vec{Z_i^b}\left(t\right) - \vec{Z_i^b}\left(t\right)\right| & r_1 > L, r_2 > E\left(t\right), \\ Game_3\colon \vec{W}_{1i}\left(t\right) \cdot \vec{Z_i^b}\left(t\right) & \\ -\vec{C}_i\left(t\right) \cdot \vec{W}_{2i}\left(t\right) \cdot \left|\vec{Z_i^b}\left(t\right) - Z_i^{\rightarrow}\left(t\right)\right| & r_1 > L, r_2 < E\left(t\right), \end{cases} \tag{2}$$

In Eq. (2), a constant number, designated to be 0.03, is denoted by the term $L$, and the vector that lies in the range of $[-C\ C]$ is indicated by $\vec{C}$. This value is used for controlling the range of activity. Hence, it shrinks gradually towards zero. Further, $r_1$ and $r_2$ denote two random numbers in the range of 0 and 1, whereas rand (l) refers to a random integer withdrawn from a uniform distribution. The weight of the starving animal is taken into account via two symbols: $\vec{W}$ and $\vec{W}$. $\vec{Z}_i^b\,(t)$ represent the optimal position of the individual at (t) iteration. $z_{\vec{i}}^{\rightarrow}\,(t)$ shows the location of $i^t$s individual. The mathematical expression of $E$ is formulated as Eq. (3).

$$E_i\,(t) = sech\,(|obj_i\,(t) - Bobj\,(t)|) \quad i \in 1, 2, 3, \ldots, K \tag{3}$$

Let $obj_i$ be the objective function value of $i^{th}$ searching agent at $t$ iteration, $Bobj$ refers to the optimal objective function that is accomplished at $t$ iteration. The overall number of individuals is represented by K. $sech\,(x)$ denotes the hyperbolic function. According to Eq. (2), an individual's behaviour can be controlled by the ranging controller $\vec{C}$ and the weights, $\vec{W}_1$ and $\vec{W}_2$. As a result, the subsequent equation explains how to refine the three parameters to enhance the search features of an individual.

For the ranging controller $\vec{C}$, the subsequent equation defines their value across the (t) iteration.

$$c_{i(t)}^{\rightarrow} = 4 \times \left(1 - \frac{iter}{\mathrm{Max}_t}\right) \times r_3 - 2 \times \left(1 - \frac{t}{\mathrm{Max}_t}\right) \tag{4}$$

In Eq. (4), $\left(1 - \frac{t}{\mathrm{Max}_t}\right)$ shows a shrinking behaviour through iteration number. $\mathrm{Max}_t$ stands for the maximal iteration count, *and* $\gamma_3$ refers to a random integer in the range of 0 and 1. For the weights $\vec{W}_1$ and $\vec{W}_2$, the subsequent formula is applied for fine-tuning the weights across the iteration number [22].

$$\vec{W}_{1i}\,(t) = \begin{cases} Hungry_i\,(t) \cdot \dfrac{K}{S\_Hungry\,(t)} \times r_4 & r_5 < L, \\ 1 & r_5 > L \end{cases} \tag{5}$$

$$\vec{W}_{2i}\,(t) = (1 - exp\,(-\,|Hungry_i\,(t) - S\_Hungry\,(t)|)) \times \gamma_6 \times 2 \tag{6}$$

Given that $Hungry_i\,(t)$ shows the hunger of an $i^{th}$ individual, $Sum\_Hungry\,(t)$ illustrates the summary of hungry feelings of every individual at (t) iteration. $r_4$, $r_5$, and $r_6$ denote the random numbers in the interval of 0 and 1. The mathematical expression of $Hungry_i\,(t)$ is determined using the following equation, Eq. (7).

$$Hungry_i\,(t) = \begin{cases} 0 & obj_i\,(t) == Bobj\,(t) \\ Hungry_i\,(t) & \\ +N\_Hungry_i\,(t) & obj_i\,(t)\,!= Bobj\,(t) \end{cases} \tag{7}$$

In this equation, $N\_Hungry_i\,(t)$ refers to a new hunger if the objective function of $i^{th}$ individuals is not equivalent to optimal fitness. Therefore, the corresponding hunger of the new individuals is different. Accordingly, the new hunger is mathematically modelled as follows.

$$N\_Hungry_i\,(t) = \begin{cases} L\_h \times (1 + r) & TH < L\_h \\ TH_i\,(t) & TH \geq L\_h. \end{cases}$$

where

$$TH_i(t) = \frac{obj_i(t) - g^*(t)}{Wobj(t) - Bobj(t)} \times r_7 \times 2 \times (U\_B - L\_B) \tag{8}$$

In Eq. (8), $Wobj\ (t)$ illustrates the worst fitness at (t) iteration. $UB$ and $LB$ symbolize the upper and lower limits of the search space, correspondingly.

The fitness Function (FF) can be defined from the classification accuracy and the number of selected features. It maximizes the classification accuracy and reduces the set size of the selected features. Thus, a subsequent fitness function can be utilized in the evaluation of individual solutions, as given in Eq. (9) below.

$$Fitness = \alpha * \ Error\ Rate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{9}$$

Here, ErrorRate refers to the classification error rate utilizing the selected features. ErrorRate can be computed as a percentage of false classifications to the number of classifications created and exhibited. It is written as a value in the range of 0 and 1. While ErrorRate can complement the classification accuracy, $\#SF$ denotes the quantity of the selected features, and $\#All\_F$ denotes the total sum of attributes in the original dataset. $\alpha$ is utilized to control the significance of subset length and classification quality. In this experiment, $\alpha$ is set to 0.9.

### *3.3 GCN-Based Intrusion Detection*

For intrusion detection and classification, the GCN model is exploited in the current study. In GCN models, the nodes are represented through pass and aggregate messages amongst the neighbouring nodes. Though various types of GCN models have been proposed earlier, the most utilized version has been used in this study, too [23]. Properly, a GCN layer is determined as given below.

$$h_i^{(l+1)} = f\left(\sum\nolimits_{j\in Ne(i)} \frac{1}{\sqrt{\widetilde{D}_{i,i}\widetilde{D}_{j,j}}} h_j^{(l)} W^{(l)}\right), \tag{10}$$

Here, $h_i^{(l)}$ represents the latent representation of the node $v_i$ from layer $l$, $Ne(i)$ signifies a group of neighbours of the node, $v_i$, and $W^l$ refers to the detailed trainable weighted matrix layer. $f$ refers to the non-linear activation function, and Rectified Linear Unit (ReLU) can be chosen as the activation function after the preceding analysis (expressed as $f_{ReLU}(\cdot)$ below). $\widetilde{D}$ represents the diagonal degree matrix of $\widetilde{A}$ and is determined as $\widetilde{D_{i,i}} = \sum_j \widetilde{A}_{i,j}$, whereas $\widetilde{A} = A + I$ refers to the adjacency matrix of the input element network $G$ with self-connection I. Regularly, GCN can be changed from matrix procedure [24]:

$$H^{(l+1)} = f_{ReLU}\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right). \tag{11}$$

For the primary layer, $H^{(0)} = X$ signifies the element matrix of input networks.

$$H^{(1)} = f_{ReLU}\left(\widetilde{A}XW^{(0)}\right). \tag{12}$$

The framework of GCN is to provide end-to-end training and incorporate task-specific loss functions. At the time of original analysis, GCN executes the semi-supervised classification tasks.

So, Cross Entropy (CE) loss is estimated with the addition of the Softmax function as the final resultant layer. The entire CE errors are estimated on a graph for every sample as labelled below.

$$\mathcal{L}_{cls} = -\sum\nolimits_{i \in L} \sum\nolimits_{c=1}^{c} Y_{ic} \log \hat{Y}_{ic} \tag{13}$$

Here, $L$ implies the group of nodes with a label, $C$ denotes the count of classes, $Y$ represents the label, and $\hat{Y} = softmax\,(H)$ indicates the forecast of GCN that passes the hidden representations from the last layer $H^{(L)}$ to the softmax function.

### *3.4 Hyperparameter Tuning Using SSO Algorithm*

Finally, the SSO technique is exploited to fine-tune the hyper-parameters related to the GCN method. SSO is inspired by the vigilant and predatory behaviours of the sparrow population [25]. Discoverer, entrant, and vigilant are the roles played by every sparrow in its population. In this work, if the fitness of an entrant is highly efficient than the discoverer, then the entrant becomes a discoverer to find their food. The updated position of the discoverer is formulated as follows.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}.\ \exp\left(-\dfrac{i}{\alpha.iter_{\max}}\right), & if\ R_2 < ST \\ X_{i,j} + Q.L, & if\ R_2 \geq ST \end{cases} \tag{14}$$

Here, $j = 1, 2, 3, d, iter_{\max}$ signifies the maximal iteration count, $t$ characterizes the current iteration, $\alpha \in [0, 1]$ denotes a random number, $X_{ij}$ denotes the location data of *i-th* sparrow in *jth* parameter, $ST \in [O.5, 1]$ signifies the safety value, $R_2 \in [0, 1]$ denotes the warning value, $L$ symbolizes a $1 \times d$ matrix whereas all the elements in this matrix are 1 and $Q$ represents a random number following a normal distribution. In this method, Entrants are a type of sparrow with low energy in the population. Sparrows with low energy find it challenging to search for food in their region. So, it flies to another location or follows the discoverer to find their food. The discoverer and the entrant replace one another based on the quantity of the stored energy. The updated position of the entrant is calculated as follows.

$$X_{i,j}^{t+1} = \begin{cases} Q.\ \exp\left(\dfrac{X_{worst} - X_{i,j}^{t}}{i^2}\right), & if\ i > \dfrac{n}{2} \\ X_P^{t+1} + \left|X_{i,j} - X_P^{t+1}\right| A^{+}.L, & otherwise \end{cases} \tag{15}$$

In Eq. (15), $X_{worst}$ denotes the worst location of the existing population, $X_P$ indicates the optimal location of the existing population, $A$ corresponds to *the* $1 \times d$ matrix, and all the elements in this matrix are random numbers in the range of 1 or $-1$, while $A^{+} = A^{T}(AA^{T})^{-1}$. If $i$ is greater than $nl2$, it implies that it is highly challenging for the sparrow with low energy to search for food. So, it should fly to other regions to search for food or towards the vicinity of the discoverer to acquire further energy. Fig. 2 demonstrates the steps involved in the SSO technique.

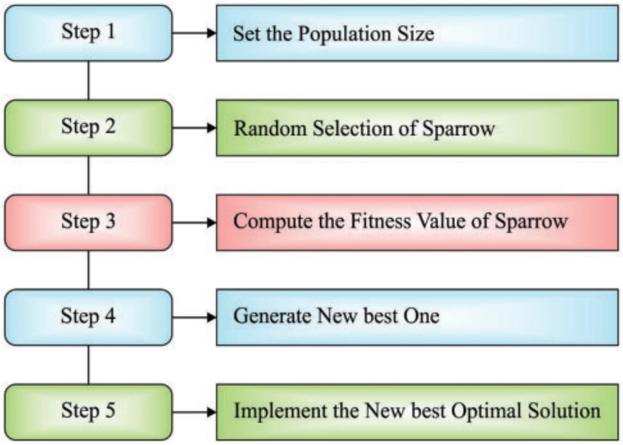
Step 1
Set the Population Size
Step 2
Random Selection of Sparrow
Step 3
Compute the Fitness Value of Sparrow
Step 4
Generate New best One
Step 5
Implement the New best Optimal Solution

**Figure 2:** Steps involved in SSO

In the sparrow population, some sparrows are named vigilant sparrows. These sparrows are generally accountable for 10%–20% of the overall population. Once it realizes the danger, the sparrow near the edge of the group moves quickly to a secure central area to avoid the danger. The sparrows at the optimum location of the population randomly walk toward other sparrows to avoid danger. The updated position of the vigilant sparrow is formulated as follows.

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta.|X_{i,j}^{t} - X_{best}^{t}|, & if\, f_i > f_g \\ X_{i,j}^{t} + K.\left(\dfrac{|X_{i,j}^{t} - X_{worst}^{t}|}{(f_i - f_w) + \varepsilon}\right), & if\, f_i = f_g \end{cases} \tag{16}$$

In Eq. (3), the optimum location of the existing population is denoted by $X_{best}$, $\beta$ represents the step size that is uniformly distributed random value with mean and variance [0, 1], $K \in [1, 1]$ denotes a random number, $f_g$ exemplifies the fitness of the present optimal sparrow, $f_i$ epitomizes the fitness of the existing sparrow, $\varepsilon$ indicates a small constant to avoid the denominator from being zero, and $f_w$ represents the fitness of the current worst sparrow. Once the fitness of the present sparrow is higher than the optimal sparrow, it implies that the sparrow is at the edge of the population and is susceptible to attack. So, such sparrows should move towards the safest place. Once the fitness of the present sparrow is equivalent to that of the optimal sparrow, it implies that the sparrow is in a safe region. Now, it moves closer to the sparrow to prevent danger.

SSO approach extracts a Fitness Function (FF) to achieve enhanced classification outcomes. It allocates a positive numeral to indicate the superior execution of a candidate solution. In this research, the reduced classification error rate is treated as the Fitness Function as given in Eq. (17).

$$fitness\,(x_i) = Classifier\ Error\ Rate\,(x_i) = \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \tag{17}$$

## 4 Experimental Validation

The proposed HGSODL-ID model was experimentally validated using the NSL-KDD dataset [26]. The dataset was generated in the year 2009 and is broadly utilized in network intrusion detection experiments. In modern literature, all the researchers employ the NSL-KDD dataset as a potential baseline dataset since it is highly helpful for researchers to compare different types of intrusion detection methodologies. Tables 1 and 2 show a detailed description of binary and multiclass classification.

**Table 1:** Details on binary classification

| NSL KDD dataset–binary classification | |
|---|---|
| Class | No. of samples |
| Normal | 77054 |
| Abnormal | 71553 |
| Total number of samples | 148607 |

**Table 2:** Details on multiclass classification

| NSL KDD dataset–multiclass classification | |
|---|---|
| Class | No. of samples |
| Normal | 77054 |
| Dos | 53475 |
| Probe | 14077 |
| R2L | 3749 |
| U2R | 252 |
| Total number of samples | 148607 |

### *4.1 Result Analysis on Binary Dataset*

Fig. 3 shows the confusion matrices created by the HGSODL-ID model on the applied binary dataset. On 70% of training (TR) data, the proposed HGSODL-ID model categorized 53,207 samples under the Normal class and 49,746 samples under the Abnormal class. Also, on 30% of testing (TS) data, the presented HGSODL-ID technique recognized 22,916 samples as Normal class and 21,214 samples as Abnormal class.

Table 3 offers the overall classification outcomes accomplished by the proposed HGSODL-ID model on the binary dataset. Fig. 4 provides the analytical results of the proposed HGSODL-ID model on 70% of TR data. The experimental outcomes imply that the proposed HGSODL-ID model achieved maximum performance in all aspects. For example, the HGSODL-ID method classified the normal class samples with an $accu_y$ of 98.97%, $prec_n$ of 98.96%, $reca_l$ of 98.98%, $F_{score}$ of 98.97%, and a $G_{mean}$ of 98.98%. Also, the proposed HGSODL-ID algorithm classified the abnormal class samples with an $accu_y$ of 98.97%, $prec_n$ of 98.68%, $reca_l$ of 99.19%, $F_{score}$ of 98.93%, and a $G_{mean}$ of 98.98%.

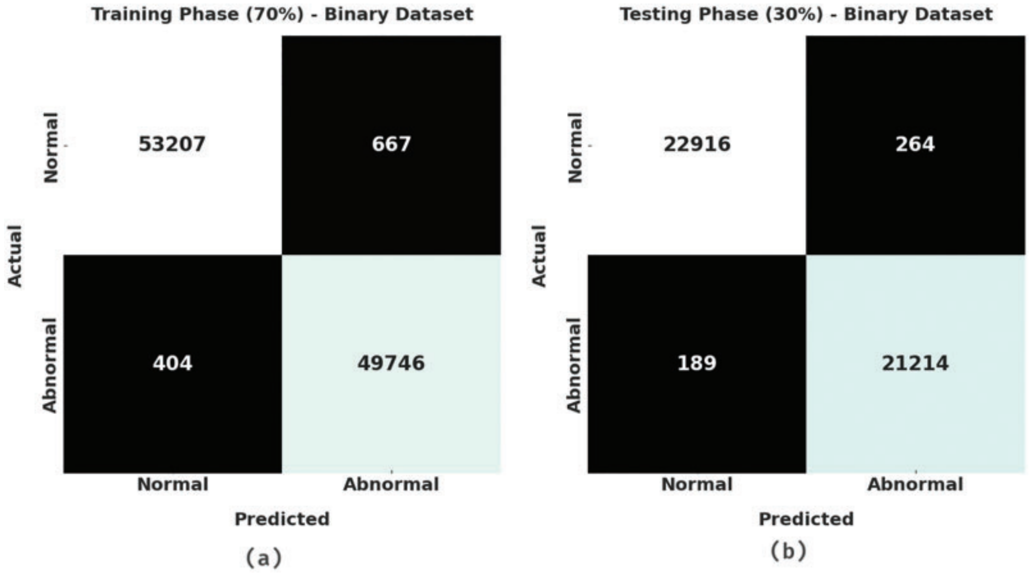

**Figure 3:** Confusion matrices of HGSODL-ID approach upon binary dataset (a) 70% of TR dataset and (b) 30% of TS dataset

**Table 3:** Results of the analysis of the HGSODL-ID approach upon binary dataset under different measures

| Labels | Accuracy | Precision | Recall | F-score | G-Mean |
|---|---|---|---|---|---|
| Training phase (70%) | | | | | |
| Normal | 98.97 | 99.25 | 98.76 | 99.00 | 98.98 |
| Abnormal | 98.97 | 98.68 | 99.19 | 98.93 | 98.98 |
| Average | 98.97 | 98.96 | 98.98 | 98.97 | 98.98 |
| Testing phase (30%) | | | | | |
| Normal | 98.98 | 99.18 | 98.86 | 99.02 | 98.99 |
| Abnormal | 98.98 | 98.77 | 99.12 | 98.94 | 98.99 |
| Average | 98.98 | 98.98 | 98.99 | 98.98 | 98.99 |

Fig. 5 portrays the results of the analysis of the HGSODL-ID approach on 30% of TS data. The experimental outcomes infer that the proposed HGSODL-ID technique obtained the maximum performance under all prospects. For example, the HGSODL-ID algorithm classified the normal class samples with an $accu_y$ of 98.98%, $prec_n$ of 99.18%, $reca_l$ of 98.86%, $F_{score}$ of 99.02%, and a $G_{mean}$ of 98.99%. Additionally, the proposed HGSODL-ID method categorized the abnormal class samples with an $accu_y$ of 98.98%, $prec_n$ of 98.77%, $reca_l$ of 99.12%, $F_{score}$ of 98.94%, and a $G_{mean}$ of 98.99%.

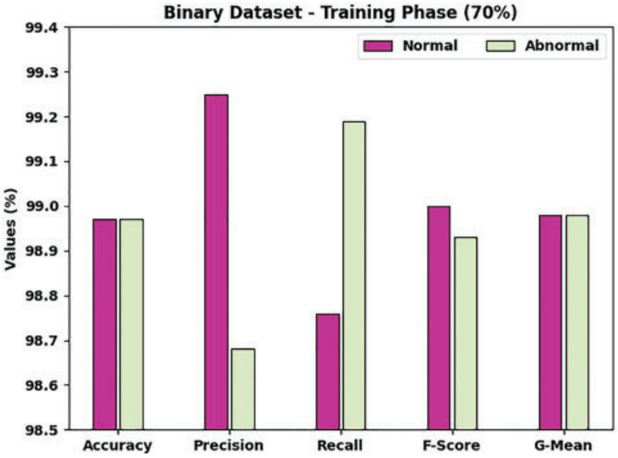

**Figure 4:** Results of the analysis of the HGSODL-ID approach on 70% of TR data in the binary dataset

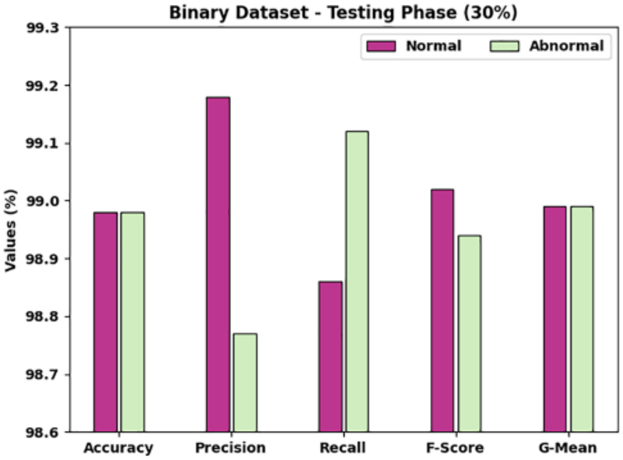

**Figure 5:** Results of the analysis of the HGSODL-ID approach on 30% of TS data in the binary dataset

Both Training Accuracy (TA) and Validation Accuracy (VA) values acquired by the proposed HGSODL-ID algorithm on the binary dataset are illustrated in Fig. 6. The experimental outcomes denote that the proposed HGSODL-ID algorithm obtained the maximal TA and VA values, while VA values were higher than TA.

Both Training Loss (TL) and Validation Loss (VL) values, accomplished by the proposed HGSODL-ID technique on a binary dataset, are established in Fig. 7. The experimental outcomes infer that the proposed HGSODL-ID method accomplished the minimal TL and VL values. In contrast, VL values were lesser than TL.

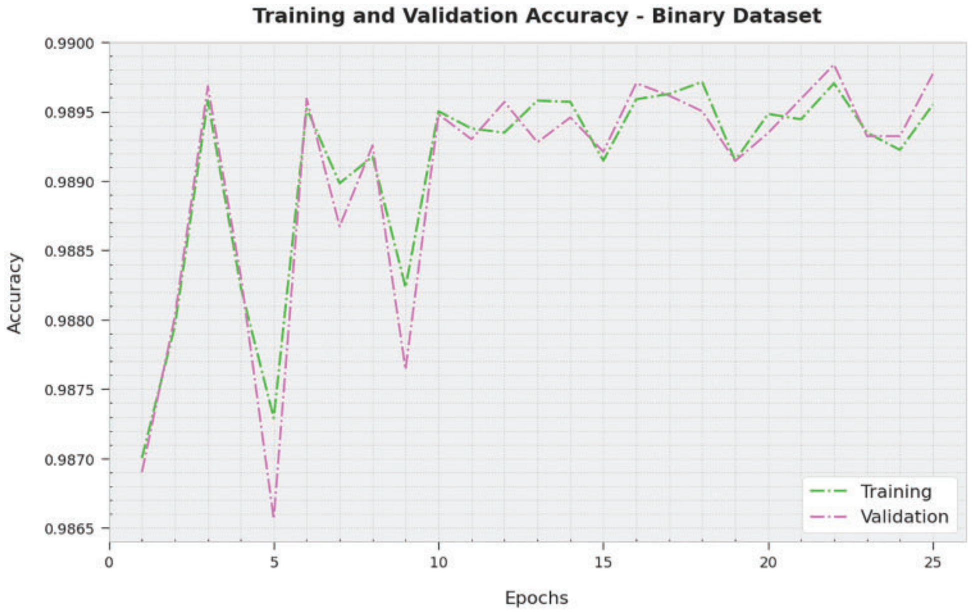

**Figure 6:** TA and VA analyses results of the HGSODL-ID approach on a binary dataset

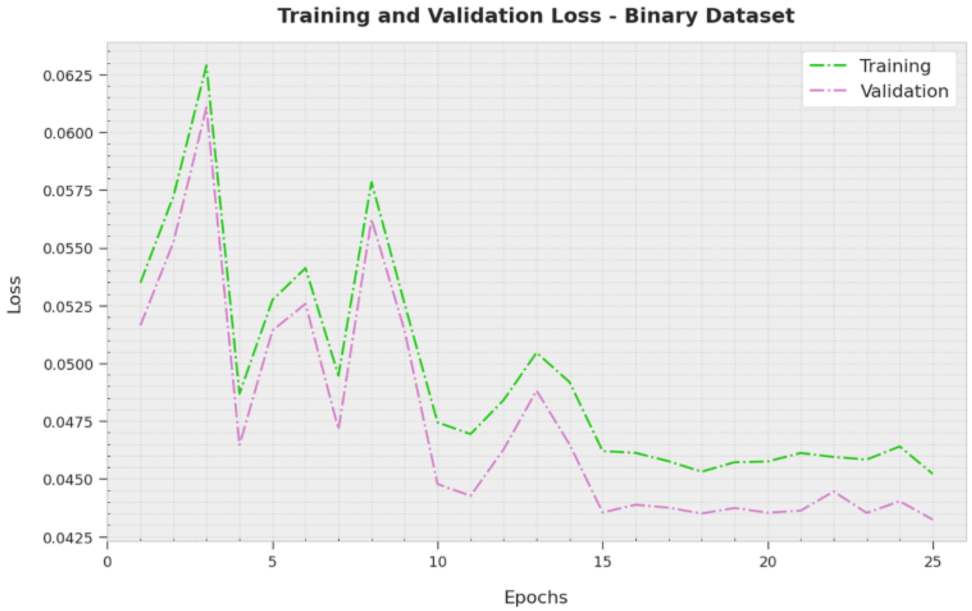

**Figure 7:** TL and VL analyses results of the HGSODL-ID approach in the binary dataset

To demonstrate the betterment of the proposed HGSODL-ID model, a detailed comparison study was conducted, and the results are shown in Table 4 and Fig. 8. The results confirmed that the AE model attained the least $accu_y$ of 82.85%. In addition, Deep Convolutional Neural Network (DCNN), Recurrent Neural Network (RNN), and Fuzziness models achieved moderately improved $accu_y$ values such as 84.44%, 83.26%, and 83.76%, respectively. Moreover, STL-IDS and Multi-CNN fusion models reported reasonable $accu_y$ values such as 85.47% and 86.42%, respectively. But, the proposed HGSODL-ID model achieved the maximum classification performance with an $accu_y$ of 98.98%.

**Table 4:** Comparative analysis results of the HGSODL-ID approach and other recent methods upon the binary dataset

| Methods | Accuracy |
|---|---|
| HGSODL-ID | 98.98 |
| DCNN | 84.44 |
| RNN | 83.26 |
| STL-IDS | 85.47 |
| Fuzziness | 83.76 |
| AE | 82.85 |
| Multi-CNN fusion | 86.42 |

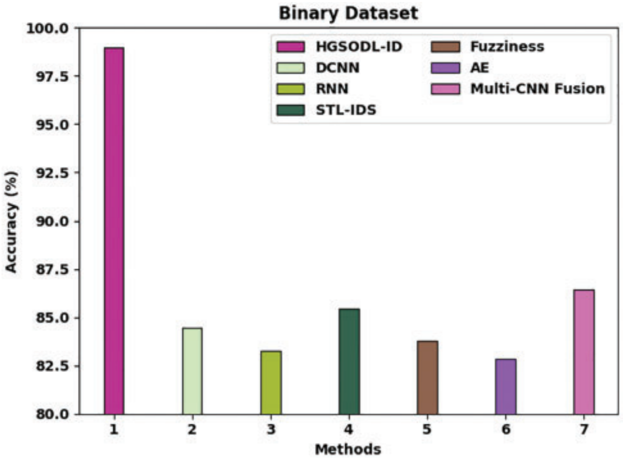

**Figure 8:** Comparative analysis results of the HGSODL-ID approach on the binary dataset

### *4.2 Result Analysis on Multiclass Dataset*

Fig. 9 portrays the confusion matrices generated by the HGSODL-ID method on the applied Multiclass dataset. On 70% of TR data, the proposed HGSODL-ID technique categorized 53,275 samples under the Normal class, 36,991 samples under the Denial of Service (DoS) class, 9,651 samples under the Probe class, 2,385 samples under the Remote-to-Local (R2L) class, and 136 samples under User to Root (U2R) class respectively. Along with that, on 30% of TS data, the presented HGSODL-ID methodology recognized 22,871 samples as Normal class, 15,837 samples as DoS class, 4,165 samples as Probe class, 999 samples as R2L class, 81 samples as U2R class.

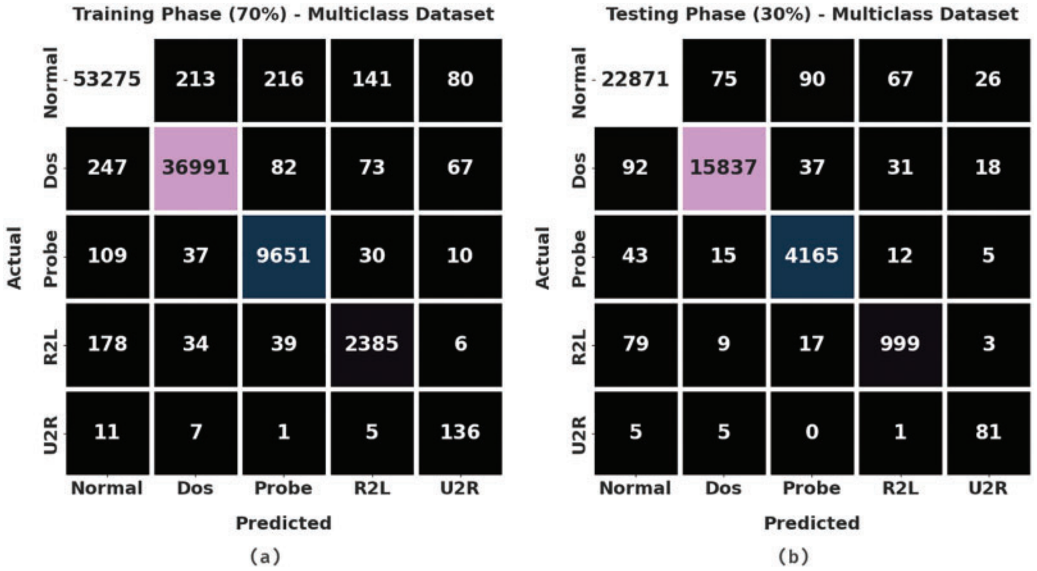

**Figure 9:** Confusion matrices of HGSODL-ID approach upon multiclass dataset (a) 70% of TR data and (b) 30% of TS data

Table 5 provides the overall classification outcomes of the proposed HGSODL-ID model on the Multiclass dataset. Fig. 10 presents the analytical results of the HGSODL-ID algorithm on 70% of TR data. The experimental outcomes infer that the proposed HGSODL-ID methodology produced the maximum performance under all aspects. For instance, the HGSODL-ID technique classified the normal class samples with an $accu_y$ of 98.85%, $prec_n$ of 98.99%, $reca_l$ of 98.79%, $F_{score}$ of 98.89%, and a $G_{mean}$ of 98.85%. Moreover, the proposed HGSODL-ID method classified the U2R class samples with an $accu_y$ of 99.82%, $prec_n$ of 45.48%, $reca_l$ of 85%, $F_{score}$ of 59.26%, and a $G_{mean}$ of 92.12%.

**Table 5:** Results of the analysis of the HGSODL-ID approach on the Multiclass dataset under different measures

| Labels | Accuracy | Precision | Recall | F-score | G-mean |
|---|---|---|---|---|---|
| Training phase (70%) | | | | | |
| Normal | 98.85 | 98.99 | 98.79 | 98.89 | 98.85 |
| Dos | 99.27 | 99.22 | 98.75 | 98.98 | 99.15 |
| Probe | 99.50 | 96.62 | 98.11 | 97.36 | 98.87 |
| R2L | 99.51 | 90.55 | 90.27 | 90.41 | 94.90 |
| U2R | 99.82 | 45.48 | 85.00 | 59.26 | 92.12 |
| Average | 99.39 | 86.17 | 94.18 | 88.98 | 96.78 |

(Continued)

**Table 5:** Continued

| Labels | Accuracy | Precision | Recall | F-score | G-mean |
|---|---|---|---|---|---|
| Testing phase (30%) | | | | | |
| Normal | 98.93 | 99.05 | 98.88 | 98.97 | 98.93 |
| Dos | 99.37 | 99.35 | 98.89 | 99.12 | 99.26 |
| Probe | 99.51 | 96.66 | 98.23 | 97.44 | 98.93 |
| R2L | 99.51 | 90.00 | 90.24 | 90.12 | 94.88 |
| U2R | 99.86 | 60.90 | 88.04 | 72.00 | 93.78 |
| Average | 99.43 | 89.19 | 94.86 | 91.53 | 97.16 |

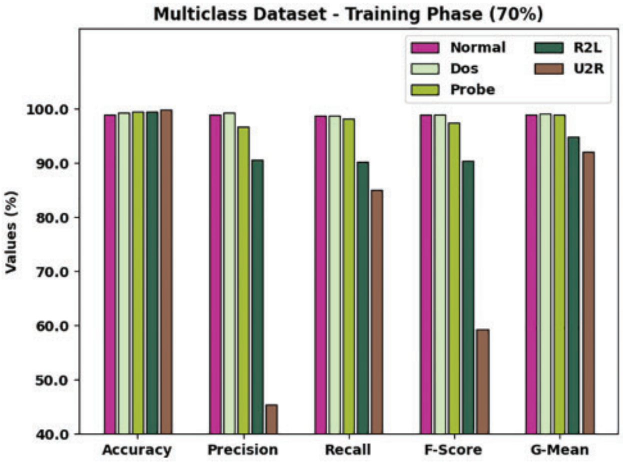

**Figure 10:** Results of the analysis of the HGSODL-ID approach on 70% of TR data in the multiclass dataset

Fig. 11 shows the analytical results of the proposed HGSODL-ID method on 30% of TS data. The experimental outcomes imply that the presented HGSODL-ID algorithm achieved the maximum performance under all aspects. For example, the HGSODL-ID approach classified the normal class samples with an $accu_y$ of 98.93%, $prec_n$ of 99.05%, $reca_l$ of 98.88%, $F_{score}$ of 98.97%, and a $G_{mean}$ of 98.93%. Furthermore, the HGSODL-ID technique classified the U2R class samples with an $accu_y$ of 99.86%, $prec_n$ of 60.90%, $reca_l$ of 88.04%, $F_{score}$ of 72%, and a $G_{mean}$ of 93.78%.

To establish the supremacy of the proposed HGSODL-ID methodology, a detailed comparative analysis was conducted and the results are shown in Table 6 and Fig. 12 [12]. The results state that Stacked Autoencoder (SAE) algorithm reached the least $accu_y$ of 79.54%. Also, Artificial Neural Network (ANN) and STL-IDS models achieved moderately improved $accu_y$ values such as 79.81% and 79.95%, correspondingly. Moreover, RNN and Multi-CNN fusion models reported reasonable $accu_y$ values such as 81.05% and 81.30% correspondingly. But, the proposed HGSODL-ID technique achieved the maximum classification performance with an $accu_y$ of 99.43%.

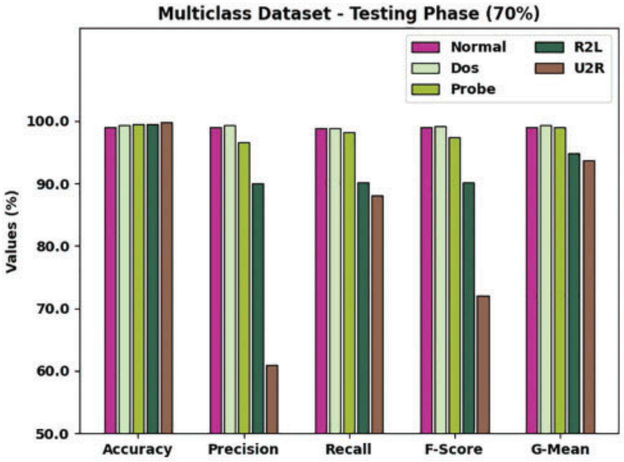

**Figure 11:** Results of the analysis of the HGSODL-ID approach on 30% of TS data in the multiclass dataset

**Table 6:** Comparative analysis results of HGSODL-ID approach and other recent methods upon multiclass dataset

| Methods | Accuracy-multiclass |
|---|---|
| HGSODL-ID | 99.43 |
| RNN model | 81.05 |
| SAE | 79.54 |
| STL-IDS | 79.95 |
| ANN | 79.81 |
| Multi-CNN fusion | 81.30 |

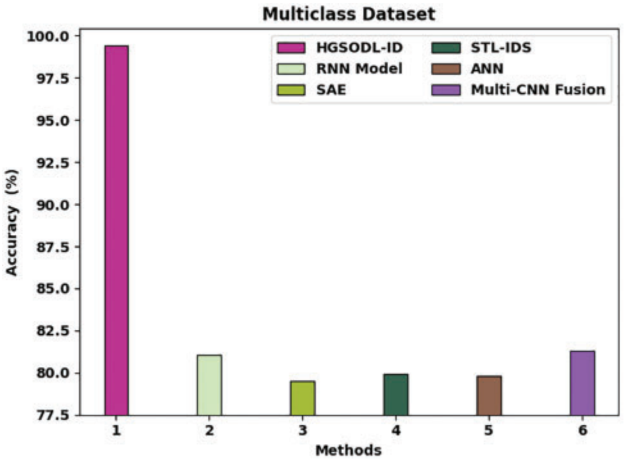

**Figure 12:** Comparative analysis results of HGSODL-ID approach on multiclass dataset

From the detailed results and discussion, it is evident that the proposed HGSODL-ID model is an excellent performer compared to other models.

## 5 Conclusion

In this study, an HGSODL-ID method has been developed for the detection and classification of intrusions in the IIoT environment. The presented HGSODL-ID technique follows a series of sub-processes namely, linear normalization, HGSO-FS-based feature selection, GCN classification, and SSO-based hyperparameter optimization. HGSO-based Feature Selection and SSO-based optimal parameter tuning enhance the detection performance of the HGSODL-ID model. The proposed HGSODL-ID method was experimentally validated with the help of benchmark datasets and the outcomes signify the supremacy of the HGSODL-ID technique over recent approaches since the method achieved the highest accuracy of 99.43%. Thus, the HGSODL-ID model can be exploited to accomplish network security in a smart factory environment. In future, the outcomes of the HGSODL-ID model can be boosted with the help of data clustering or outlier removal approaches.

**Funding Statement:** Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4340237DSR44. The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program Grant Code (NU/RG/SERC/11/4).

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

## References

[1] S. Latif, Z. Idrees, Z. Zou and J. Ahmad, "DRaNN: A deep random neural network model for intrusion detection in industrial iot," in *2020 Int. Conf. on UK-China Emerging Technologies (UCET)*, Glasgow, United Kingdom, pp. 1–4, 2020.

[2] T. Hasan, J. Malik, I. Bibi, W. U. Khan, F. N. A. Wesabi *et al.,* "Securing industrial internet of things against botnet attacks using hybrid deep learning approach," *IEEE Transactions on Network Science and Engineering*, pp. 1, 2022. https://doi.org/10.1109/TNSE.2022.3168533

[3] S. Tharewal, M. W. Ashfaque, S. S. Banu, P. Uma, S. M. Hassen *et al.,* "Intrusion detection system for industrial internet of things based on deep reinforcement learning," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–8, 2022.

[4] I. Essop, J. C. Ribeiro, M. Papaioannou, G. Zachos, G. Mantas *et al.,* "Generating datasets for anomaly-based intrusion detection systems in iot and industrial iot networks," *Sensors*, vol. 21, no. 4, pp. 1528, 2021.

[5] M. A. Hawawreh, N. Moustafa and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.

[6] A. A. Albraikan, S. B. Haj Hassine, S. M. Fati, F. N. Al-Wesabi, A. M. Hilal *et al.,* "Optimal deep learning-based cyberattack detection and classification technique on social networks," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 907–923, 2022.

[7] P. L. S. Jayalaxmi, R. Saha, G. Kumar and T. -H. Kim, "Machine and deep learning amalgamation for feature extraction in industrial internet-of-things," *Computers & Electrical Engineering*, vol. 97, pp. 107610, 2022.

[8] A. M. Hilal, J. S. Alzahrani, I. Abunadi, N. Nemri, F. N. Al-Wesabi *et al.,* "Intelligent deep learning model for privacy preserving iiot on 6g environment," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 333–348, 2022.

[9] M. A. Alohali, F. N. Al-Wesabi, A. M. Hilal, S. Goel, D. Gupta *et al.,* "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment," *Cognitive Neurodynamics*, 2022. https://doi.org/10.1007/s11571-022-09780-8

[10] E. Gyamfi and A. Jurcut, "Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets," *Sensors*, vol. 22, no. 10, pp. 3744, 2022.

[11] A. M. Hilal, M. A. Alohali, F. N. Al-Wesabi, N. Nemri, J. Hasan *et al.,* "Enhancing quality of experience in mobile edge computing using deep learning based data offloading and cyberattack detection technique," *Cluster Computing*, 2021. https://doi.org/10.1007/s10586-021-03401-5

[12] A. Fatani, A. Dahou, M. A. A. Al-Qaness, S. Lu and M. A. Abd Elaziz, "Advanced feature extraction and selection approach using deep learning and aquila optimizer for iot intrusion detection system," *Sensors*, vol. 22, no. 1, pp. 140, 2021.

[13] G. Altan, "SecureDeepNet-IoT: A deep learning application for invasion detection in industrial internet of things sensing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, 2021.

[14] S. T. Park, G. Li and J. C. Hong, "A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1405–1412, 2020.

[15] J. B. Awotunde, C. Chakraborty and A. E. Adeniyi, "Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–17, 2021.

[16] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng *et al.,* "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, vol. 154, pp. 107450, 2020.

[17] M. A. Basset, V. Chang, H. Hawash, R. K. Chakrabortty and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7704–7715, 2021.

[18] M. Al-Hawawreh, E. Sitnikova and F. den Hartog, "An efficient intrusion detection model for edge system in brownfield industrial internet of things," in *Proc. of the 3rd Int. Conf. on Big Data and Internet of Things*, Melbourn VIC Australia, pp. 83–87, 2019.

[19] E. Gyamfi and A. D. Jurcut, "Online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM," *IEEE Internet Things Journal*, pp. 1, 2022. https://doi.org/10.1109/JIOT.2022.3172393

[20] A. Fatani, A. Dahou, M. A. A. Al-qaness, S. Lu and M. A. A. Elaziz, "Advanced feature extraction and selection approach using deep learning and aquila optimizer for iot intrusion detection system," *Sensors*, vol. 22, no. 1, pp. 140, 2021.

[21] Y. Yang, H. Chen, A. A. Heidari and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, pp. 114864, 2021.

[22] H. Nguyen and X. -N. Bui, "A hunger games search optimization-based artificial neural network for predicting ground vibration intensity induced by mine blasting," *Natural Resources Research*, vol. 30, no. 5, pp. 3865–3880, 2021.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

[24] Y. Pei, T. Huang, W. van Ipenburg and M. Pechenizkiy, "ResGCN: Attention-based deep residual modeling for anomaly detection on attributed networks," *Machine Learning*, vol. 111, no. 2, pp. 519–541, 2022.

[25] Z. Wang, X. Huang and D. Zhu, “A multistrategy-integrated learning sparrow search algorithm and optimization of engineering problems,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–21, 2022.

[26] S. K. Sahu, S. Sarangi and S. K. Jena, “A detail analysis on intrusion detection datasets,” in *2014 IEEE Int. Advance Computing Conf. (IACC)*, Gurgaon, India, pp. 1348–1353, 2014.