energies
MDPI

Article

A Classy Multifacet Clustering and Fused Optimization Based Classification Methodologies for SCADA Security

Alaa O. Khadidos [1,*], Hariprasath Manoharan [2], Shitharth Selvarajan [3,*], Adil O. Khadidos [4], Khaled H. Alyoubi [1] and Ayman Yafoz [1]

[1] Department of Information Systems, Faculty of Computing and Information Systems, King Abdulaziz University, Jeddah 22254, Saudi Arabia; kalyoubi@kau.edu.sa (K.H.A.); ayafoz@kau.edu.sa (A.Y.)
[2] Department of Electronics and Communication Engineering, Panimalar Institute of Technology, Poonamallee, Chennai 600123, India; hari13prasath@gmail.com
[3] Department of Computer Science & Engineering, Kebri Dehar University, Kebri Dehar P.O. Box 250, Ethiopia
[4] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 22254, Saudi Arabia; akhadidos@kau.edu.sa
* Correspondence: aokhadidos@kau.edu.sa (A.O.K.); shitharth.it@gmail.com (S.S.)

Abstract: Detecting intrusions from the supervisory control and data acquisition (SCADA) systems is one of the most essential and challenging processes in recent times. Most of the conventional works aim to develop an efficient intrusion detection system (IDS) framework for increasing the security of SCADA against networking attacks. Nonetheless, it faces the problems of complexity in classification, requiring more time for training and testing, as well as increased misprediction results and error outputs. Hence, this research work intends to develop a novel IDS framework by implementing a combination of methodologies, such as clustering, optimization, and classification. The most popular and extensively utilized SCADA attacking datasets are taken for this system's proposed IDS framework implementation and validation. The main contribution of this work is to accurately detect the intrusions from the given SCADA datasets with minimized computational operations and increased accuracy of classification. Additionally the proposed work aims to develop a simple and efficient classification technique for improving the security of SCADA systems. Initially, the dataset preprocessing and clustering processes were performed using the multifacet data clustering model (MDCM) in order to simplify the classification process. Then, the hybrid gradient descent spider monkey optimization (GDSMO) mechanism is implemented for selecting the optimal parameters from the clustered datasets, based on the global best solution. The main purpose of using the optimization methodology is to train the classifier with the optimized features to increase accuracy and reduce processing time. Moreover, the deep sequential long short term memory (DS-LSTM) is employed to identify the intrusions from the clustered datasets with efficient data model training. Finally, the proposed optimization-based classification methodology's performance and results are validated and compared using various evaluation metrics.

Keywords: supervisory control and data acquisition (SCADA); intrusion detection system (IDS); multifacet data clustering model (MDCM); artificial intelligence; gradient descent spider monkey optimization (GDSMO); deep sequential long short term memory (DS-LSTM)

check for updates

Citation: Khadidos, A.O.; Manoharan, H.; Selvarajan, S.; Khadidos, A.O.; Alyoubi, K.H.; Yafoz, A. A Classy Multifacet Clustering and Fused Optimization Based Classification Methodologies for SCADA Security. *Energies* **2022**, *15*, 3624. https://doi.org/10.3390/en15103624

Academic Editors: Shriram Srinivasarangan Rangarajan, Tomonobu Senjyu, E. Randolph Collins and Umashankar Subramaniam

Received: 20 April 2022
Accepted: 13 May 2022
Published: 15 May 2022

1. Introduction

Supervisory control and data acquisition (SCADA) [1,2] is a software application system extensively utilized in many industrial sectors to monitor, control, and analyze manufacturing units. Due to its increased efficiency and performance, SCADA is utilized worldwide in different fields and industries to facilitate proper industrial operations. Additionally, SCADA systems [3–5] are mainly used to monitor, control, and automate the industrial processes by collecting the data from remote units and equipment, such as human machine interfaces (HMI), programmable logic controllers (PLC), and remote terminal units (RTU). However, providing security to SCADA against network attacks [6,7] is one of the most challenging and difficult tasks in the current era due to the rapid increase in attacks. Therefore, to safeguard SCADA systems, the intrusion detection system (IDS) has been developed to help identify harmful intrusions or attacks against networking operations [8,9]. Additionally, it directs the attacking alerts to the network administrators in order to ensure the security of systems. Typically, the IDS is considered as the most suitable and alternative security approach, and it is highly preferred by many researchers [10]. In this framework, the software program can be used to monitor and detect malicious activities, such as breaking of protocols, interrupting the network communication/data transmission, and data theft. Moreover, it is more suitable [11,12] for detecting both the known and unknown attacks in the network created by internal/external attackers. However, most of the conventional IDS approaches are not able to handle the complex nature of cyber-attacks. Hence, ensuring the security of SCADA systems remains a challenging process.

Some of the existing works [13,14] aim to incorporate the clustering, optimization, and classification methodologies with the IDS framework to resolve this problem. Recently, machine learning and deep learning techniques are increasingly utilized by many researchers to detect network intrusions by extracting dataset features [15]. These include the mechanisms [16,17] of the naïve Bayes (NB), the support vector machine (SVM), logistic regression (LR), linear discriminant analysis (LDA), the decision tree (DT), the random forest (RF), the multilayer perceptron (MLP), ensemble learning (EL), the deep neural network (DNN), the recurrent neural network (RNN), and the convolutional neural network (CNN). Yet, it faces problems [18,19] and challenges related to complex computational operations, increased time consumption for training and testing, and a high misclassification and error rate. Hence, the proposed work intends to implement an intelligent and hybrid IDS framework using sophisticated optimization and classification methodologies for spotting intrusions from SCADA IDS datasets. The novelty of this system is to group the attributes into the form of clusters before selecting the optimal number of features for training the classifier. The main contribution of the proposed work is to detect intrusions from the given SCADA datasets with reduced computational complexity and increased accuracy. For this purpose, a combination of methodologies are used to construct a simple and efficient intrusion detection framework for ensuring the security of SCADA systems. Additionally, the proposed objective is to implement intelligent and advanced clustering, optimization, and classification methodologies for developing the proposed security framework.

The primary objectives of the research methodology are as follows:

- To preprocess and normalize the given IDS dataset by grouping the attributes into the form of clusters, the multifacet data clustering model (MDCM) is implemented, which helps to simplify the process of classification.
- To optimally select the features for increasing the efficiency of classifier training, the gradient descent spider monkey optimization (GDSMO) mechanism is utilized, which minimizes the time of processing and increases the convergence rate.
- To exactly spot the intrusions from the clustered datasets based on the optimal set of features, the deep sequential long short term memory (DS-LSTM) technique is employed.
- To assess the performance of the proposed GDSMO-DSLSTM-based IDS framework, various evaluation measures have been utilized, and the obtained results are compared with other recent IDS approaches.

The remaining units of this paper are segregated into the following: some of the conventional clustering, optimization, and classification techniques used to increase SCADA systems' security are reviewed with their advantages and disadvantages in Section 2. The working methodology of the proposed system is illustrated with its overall flow and algorithmic representations in Section 3. The performance analysis of the proposed IDS framework is validated and compared by using various evaluation metrics in Section 4. Finally, the overall paper is summarized with its future scope in Section 5.

2. Related Works

This section reviews some of the conventional approaches used for developing an IDS in SCADA systems. Additionally, it investigates the benefits and limitations of each mechanism based on its characteristics and working operations.

Ref. [20] implemented a deep learning model for detecting intrusions in SCADA systems, where the network-based cyber-attack primitives were highly concentrated. Additionally, it mainly aims to extract the features and salient temporal patterns of individual packets by using the convolutional neural network (CNN) algorithm. [21] presented a comprehensive review of various IDS methodologies for increasing the security of SCADA systems. The primary factor of this work was to analyze the different types of methodologies used for detecting the attacks, which include the following types: intrusion detection technologies, intrusion detection methodologies, and intrusion detection approaches. Moreover, an effective IDS should satisfy the following constraints:

- Accurate detection
- Improved system reliability
- Reduced false positives
- Ability to handle large dimensional datasets
- Fast processing

Ref. [22] implemented a hybrid multilevel (HML) IDS mechanism incorporated with the nearest neighbor rule algorithm for detecting industrial attacks. The main purpose of this work was to exactly detect the anomalies with reduced false positives and an increased detection rate. Here, three different feature selection mechanisms have been analyzed and compared for improving the dimensionality of features. In addition to that, the Bloom filtering approach was utilized for categorizing the normal network patterns and anomalies by constructing the hash lookup table. The key advantages of this work were optimal performance and minimal resource consumption. Yet, it faced the problems of complex analysis, as well as the inability to handle different types of attacks. Ref. [23] developed an anomaly-based IDS (Ab-IDS) for spotting cyber-attacks in SCADA systems. This work mainly aims to identify malicious packets in the network with reduced system disturbances and network traffic. For validating the performance of this approach, two different IDS security tools, such as Snort and Bro, have been utilized.

Ref. [24] employed a long short term memory (LSTM) classification technique for detecting intrusions in the SCADA system. This work mainly aimed to identify temporal uncorrelated attacks by analyzing the specific features from the given dataset. It includes nearly 19 different types of features, such as port number, sequence number, traffic type, threshold value, speed, register data, etc. Typically, LSTM is a kind of deep learning-based classification technique that helps to predict accurate labels for given problems. Here, the many-to-many (MTM) and many-to-one (MTO) architectures have been developed for improving the performance of attack detection. Still, it has limits, such as the problems of increased time consumption for forming the hidden layers, and complexity in handling the large data. Ref. [25] presented a novel intrusion detection framework for identifying malicious activities in SCADA systems. This paper analyzed the performance and efficiency of two different and popular IDS technologies, such as Snort and Suricata, for categorizing the types of intrusions. Moreover, it investigated some of the security challenges in SCADA systems, which includes the following: lack of security in communication, inefficient data training, authentication, and controlling.

Ref. [26] deployed an auto-encoder-based network IDS for locating critical attacks in SCADA systems based on the 17 distinct data features. Here, the distributed network protocol 3 (DNP3) has been utilized for ensuring reliable communication in the network. In addition to this, hyper-parameter optimization was performed in this work for training the auto-encoder based on the hyper-parameters. Additionally, the effectiveness of this model has been validated and compared based on the measures of accuracy, precision, recall, and false positives [15]. The benefits of this work were minimized error value and processing time due to the hyper-parameter tuning. Ref. [27] employed a feed-forward neural network (FNN) mechanism for identifying correlated and uncorrelated attacks with ensured performance outcomes. Here, the omni attack detector has been developed for distinguishing the different types of attacks. The detection performance of this work could be enhanced based on the features of communication traffic and threshold value. Yet, it has the drawbacks of reduced scalability, reliability, and real-time monitoring was not possible in this system. Ref. [28] presented a comprehensive analysis of various machine learning techniques used for detecting intrusions in SCADA networks, which include the mechanisms of the support vector machine (SVM), the random forest (RF), the J48 classifier, the naïve Bayes (NB), and the decision tree. The key factor of this work was to select the most suitable technique used for increasing the performance of IDS. Based on this study, it was identified that the random forest classifier technique outperforms the other techniques with reduced error rate and false positives.

Ref. [29] implemented an elephant herding optimization (EHO)-based recurrent neural network (RNN) classification technique for detecting intrusions in IoT-SCADA systems. Here, the Caesar ciphering model integrated with the elliptic curve cryptography mechanism was utilized for improving the security level of SCADA systems. The primary advantages of this work were increased detection accuracy, security, and reduced training time. Ref. [30] introduced a new SCADA framework for industrial applications with ensured security and reliable data communication. This work mainly intends to analyze the major risk factors that could affect the performance of SCADA systems. Here, some of the common characteristics, such as data base injections, communication, and prioritization of tasks have been investigated for improving the performance of SCADA systems. Moreover, the detailed vulnerability assessment test has been conducted for validating the detection efficiency of intrusion detection and classification. Ref. [31] examined the performance of various machine learning classification approaches, such as SVM, RF, DT, logistic regression, NB, and KNN for developing an efficient SCADA-IDS. For this analysis, the online real-time traffic data has been utilized, while the training and testing assessments were performed for attack identification and categorization.

Ref. [32] introduced a new framework named as the Dnp3 intrusion detection prevention system (DIDEROT) for increasing the security of SCADA systems. Here, the attack detection was performed based on the analysis of network topology, and the developed framework was used to mitigate both the anomalies and DNP3 cyber-attacks. Moreover, it includes the modules of preprocessing, training and prediction, in which the data preprocessing could be performed based upon min-max scaling, normalization, and robust scaling. After that, the machine learning classification methodology was implemented to train the preprocessed data to detect the anomalies. The key benefit of this work was that it was capable of operating in both NIDS and HIDS. Ref. [33] developed a biased intrusion scheme for increasing the security of SCADA systems, which comprises the phases of optimization, classification, and security. Here, the modified GWO technique was implemented to analyze the features of data in order to sort the malfunctions. Then, the entropy-based ELM technique was utilized to detect the intruders based on the parameters of date, time, and file location. Finally, a hybrid ECC technique was employed to select the trusted routing path [34] for securing the information against the attackers. Ref. [35] aimed to identify the potential breaches and vulnerabilities in the SCADA systems by providing some recommendations to ensure the security of network. Here, the different types of overflow vulnerabilities, such as stack-based, multiple buffer, heap-based, multiple heap-based, multiple stack-based, and buffer overflows could be investigated with the strategy of attacks and interruptions. Ref. [36] employed a chicken swarm optimization-based deep CNN technique for detecting cracks on the concrete structures. The main purpose of this work was to analyze the structural condition of concretes for identifying the damages of cracks, spalling, exposure, and rebar buckling. Here, group statistical evaluation metrics have been used to validate the results of this scheme. Ref. [37] utilized a GA-based CNN technique for detecting the concrete cracks with increased accuracy. Here, the hyper-parameter optimization [38] could be performed for tuning the parameters of learning rate, number of layers, and optimization function.

According to this review, it is studied that the existing works are highly concentrating on developing the IDS frameworks with the data clustering, optimization, and classification approaches. Yet, this approach faces the problems and challenges related to the following:

- Inability in handling large datasets
- High false positives and error outputs
- Misclassification results
- Requires high time consumption for training data
- Follows complex computational operations for classification

Hence, the proposed work aims to develop an advanced and intelligent optimization -based classification methodology for developing the intrusion detection framework in SCADA systems.

3. Proposed Methodology

This section presents the working methodology of the proposed IDS system used for detecting intrusions from the SCADA systems. The primary objective of this work is to accurately spot the intrusions from the IDS datasets by using a combination of clustering, optimization, and classification methodologies with reduced computational complexity and time consumption. For accomplishing this process, a multifacet data clustering model (MDCM), gradient descent spider monkey optimization (GDSMO), and deep sequential long short term memory (DS-LSTM) have been implemented. The novel contribution of the proposed system is to select the optimal features from the clustered dataset based on the best fitness value for detecting and categorizing the type of intrusions with an efficient data training model. Here, the SCADA IDS datasets have been taken as the inputs for processing, which comprises some irrelevant attribute information, random values, and a missing field of attributes. Hence, it must be preprocessed and clustered to improve the quality of input datasets, because the unbalanced dataset can affect the performance of IDS with increased misclassification results and error values. So, the proposed work intends to utilize the MDCM technique for normalizing and clustering the data attributes of the input dataset, which helps to improve the efficiency and accuracy of classification. After that, the GDSMO mechanism is implemented for optimally selecting the most-suited features from the clustered dataset, based on the best fitness value. Here, the main advantages of using the GDMO technique are as follows: it efficiently identified the best global optimal solution with minimum iterations, increased convergence rate, and was fast in processing. Moreover, the DS-LSTM mechanism is employed to detect the intrusions from the desired datasets by using the set of optimal features. This is because it supports the aim of efficiently training the model of classifier with reduced time consumption and increased accuracy. Finally, the classifier produces the predicted label as whether normal or intrusion.

The working flow and methodology of the proposed IDS in SCADA systems is shown in Figure 1, which involves the following modules of operations:

- Data preprocessing and clustering
- Segmentation
- Feature Optimization
- Attack Prediction

3.1. Data Preprocessing and Clustering

At first, the input dataset preprocessing and normalization processes have been performed for balancing the attributes by filling the missing values, and eliminating the irrelevant information and random values. Additionally, dataset clustering is one of the most essential operation that needs to be accomplished for segmenting the dataset into the group of attribute information in the form of clusters. This is because the large and unbalanced datasets are highly difficult to process, and they also affects performance of classification with increased error values and false positives. Hence, this work aims to implement an advanced clustering technique, named as the multifacet data clustering model (MDCM), for normalizing and clustering the original input datasets, which helps to improve the performance of the classifier. The key factors of using this technique are reduced detection time, increased speed of processing, and classifier accuracy. This stage includes the following stages:

- Attribute normalization
- Distance computation
- Clustering

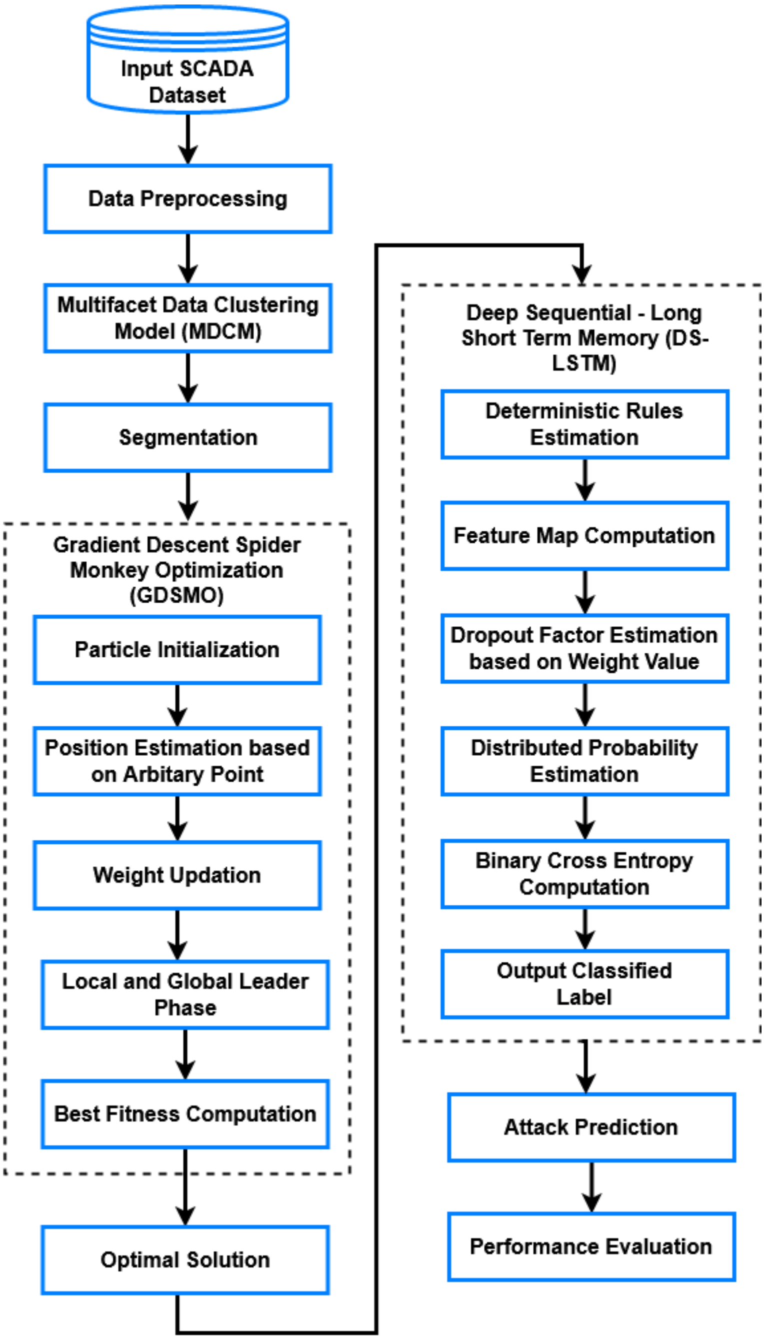

Input SCADA Dataset
Data Preprocessing
Multifacet Data Clustering Model (MDCM)
Segmentation
Gradient Descent Spider Monkey Optimization (GDSMO)
Particle Initialization
Position Estimation based on Arbitary Point
Weight Updation
Local and Global Leader Phase
Best Fitness Computation
Optimal Solution
Deep Sequential - Long Short Term Memory (DS-LSTM)
Deterministic Rules Estimation
Feature Map Computation
Dropout Factor Estimation based on Weight Value
Distributed Probability Estimation
Binary Cross Entropy Computation
Output Classified Label
Attack Prediction
Performance Evaluation

Figure 1. Working flow of the proposed methodology.

Here, the attribute normalization is mainly performed for standardizing the data values by extracting the relevant features, where the data is normalized between the values of 0 to 1 as shown in the following equation:

$$f'_v = \frac{f_v - Min\ (DS)}{Max\ (DS) - Min\ (DS)} \tag{1}$$

where, f'_v indicates the normalized feature value, $Min\ (DS)$ and $Max\ (DS)$ denote the minimum and maximum values of the dataset DS, respectively, and the feature value of $f_v \in DS$. Then, the distance computation is performed to estimate the similarity between the multiple features of the data, which is computed according to the minimum distance and increased similarity value. Consider the input dataset having two objects with N number of attributes as $DS_i = \{f_{v1}, f_{v2} \ldots f_{viN}\}$. and $DS_j = \{f_{v1}, f_{v2} \ldots f_{vjN}\}$. After that, the correlation between the data is estimated based on the formation of a covariance matrix as illustrated in the equation below:

$$d(DS_i, DS_j) = \sqrt{(DS_i - DS_i)^S CM^{-1}(DS_i - DS_j)} \tag{2}$$

where, $d(.)$ indicates the distance function, and CM is the generated covariance matrix. Moreover, the estimated distance function is mainly used to compute the similarity of multi-features in the dataset. Consequently, the symmetry similarity matrix $m \times m$ has been constructed according to the closeness of data objects as illustrated in the equation below:

$$\begin{pmatrix} 0 & d(DS_1, DS_2) & \ldots & d(DS_1, DS_n) \\ d(DS_2, DS_1) & 0 & \ldots & d(DS_2, DS_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(DS_n, DS_1) & d(DS_n, DS_2) & \ldots & 0 \end{pmatrix} \tag{3}$$

Furthermore, the best clustering effects has been obtained by using the following Equation (4):

$$\delta = \sum_{j=1}^{N} \sum_{DS_i \in C_j} ||DS_i - C_j||^2 \tag{4}$$

where, δ indicates the clustering result, and C_j denotes the center of the j-th cluster. Based on the minimum distance value, the clustering dataset has been generated, which is used for further operations, such as optimization and classification.

3.2. Gradient Descent Spider Monkey Optimization (GDSMO)

After preprocessing, the optimal number of features are selected from the clustered dataset based on global fitness function by using the proposed hybrid Gradient Descent Spider Monkey Optimization (GDSMO). The conventional SMO technique can easily fall into the problem of local optimum, hence it could not be suitable for all kinds of applications. Hence, the proposed work intends to incorporate the gradient descent (GD) with SMO technique, which efficiently avoids the local optimum problem by adding the fraction of past weight update with the current weight update value. Additionally, it acts like a simulated annealing algorithm, where the randomness is hosted to avoid the local minimum of optimization. In this technique, the parameters are initialized with the random values, and the derivatives are computed to adjust the weight value according to the objective function.

The main purpose of using this technique is to select the best features with a reduced number of iterations, increased convergence rate, and speed of processing. Additionally, it is a technique inspired by a stochastic optimization mechanism, which helps to efficiently reduce the learning time of the classifier [39]. Typically, the increased number of features can degrade the performance of classification with an increased time consumption and misprediction rate. Hence, it is most essential to optimally select the best suited features in order to train the data model of a classifier for intrusion identification and classification. Here, the parameter tuning is performed for simplifying the process of classification, due to the fact that it is more suitable for solving the complex multi-objective optimization problems. In this technique, the local iterative search is enabled for calculating the functions having a local minimum. Consider that the multivariate function $M(x)$ is distinctive from the neighboring points k, and that $M(k)$ is decreased with the negative gradient of $(k, G(k))$, denoted as the gradient descent. Then, the next position P of the gradient corresponding to the current position k is illustrated as follows:

$$P = k - \omega \nabla M(k) \tag{5}$$

where, ω indicates the weight factor. The function $M(k) > M(P)$ must be satisfied to confirm the sufficient level of ω. Consequently, the sequence of attributes $s_0,\ s_1, s_2 \ldots$ and $t_0,\ t_1, t_2 \ldots$ are considered with an arbitrary point s_0, and the local minimum value is computed as follows:

$$M(t_i) = \rho(s_i - t_i)^2 + \delta(t_i - t_{i+1})^2 + \delta(t_i - t_{i-1})^2 \tag{6}$$

$$t'_i = t_i + 2\rho(s_i - t_i) + 2\delta(t_{i+1} - t_i - 2t_i)^2 \tag{7}$$

Based on the step function, the expected local point is optimally identified with improved convergence. This optimization algorithm performs the following operations for computing the best fitness value:

- Initialization
- Local Leader Selection
- Global Leader Selection
- Learning module
- Decision module

During initialization, there are E number of spider monkeys which have been initialized, in which each monkey has the set of the G dimensional vector as $B_{ij}(i = 1, 2, 3 \ldots E)$, where B_{ij} indicates the i-th spider monkey B at the j-th direction. This is represented as follows:

$$B_{ij} = B_{mnj} + rand\ (0, 1)\left(B_{mxj} - B_{mnj}\right) \tag{8}$$

where, B_{mnj} and B_{mxj} are the minimum and maximum limits of the spider monkey B_{ij}, and the function rand (0, 1) indicates the random value lies in the range of 0 to 1. After initialization, the local leader is selected from the group of local members, and the fitness is computed according to its new position. If the estimated fitness value is greater than the new fitness value, the spider monkeys have updated their position as shown in the equation below:

$$B_{hij} = B_{ij} + rand\ (0, 1)\left(LP_{vj} - B_{ij}\right) + rand(-1, 1)\left(B_{rj} - B_{ij}\right) \tag{9}$$

where, B_{hij} is the new position of the spider monkey, LP_{vj} indicates the v-th local group leader with dimension j, and B_{rj} denotes the random r-th spider monkey with dimension j, $r \neq i$. Subsequently, the global leader is elected based on the experience, and during this stage, all spider monkeys have to update their positions. Then, the experience of both local and global leader members are determined as follows:

$$B_{hij} = B_{ij} + rand\ (0,\ 1)\left(GP_j - B_{ij}\right) + rand(-1, 1)\left(B_{rj} - B_{ij}\right) \tag{10}$$

where, GP_j indicates the global leader with dimension j and random index of $j \in \{1, 2 \ldots d_n\}$. Then, the positions of all spider monkeys (B_i) have been updated according to the probability value of Pb_i. This value can be determined with respect to the fitness value and, based on this, the best global leader candidate is selected using the probability value as shown below:

$$Pb_{ij} = \frac{F_i}{\sum_{i=1}^{n} F_i} \tag{11}$$

where, Pb_{ij} indicates the estimated probability function, and F_i is the fitness value of the i-th spider monkey. Furthermore, the learning phase has been executed with the local and global leaders. During this process, the spider monkey having the highest fitness value is considered as the global leader of all spider monkeys, and its position does not update. Similar to that, the local leader has been selected from each group of members, and its position is also does not update. During the decision making module, the group members have to update their positions once the local limit reaches the threshold value, as shown in the equation below:

$$B_{hij} = B_{ij} + T(0,1) \times \left(GP_j - B_{ij}\right) + T(0,1) \times \left(B_{ij} - LP_{vj}\right) \tag{12}$$

Similar to that, the global leader could split the population into small number groups, until it reached the maximum number of splits. If its position is not updated, all groups are integrated into a single group. Based on the optimal solution, the final best subset of features have been selected for improving the accuracy of classification. These selected features are further utilized for training the classifier that helps to increase the overall accuracy of intrusion detection and classification system. The algorithmic procedure of the proposed IDS is presented in Algorithm 1.

Algorithm 1 Gradient Descent Spider Monkey Optimization (GDSMO)

```
Input:      Initial set of population $s_i(a \le i \le m)$, transaction probability $\tau$, and switching probability $\alpha_p$;
Output :    Best optimal solution $Opt_{a(i)}$;
Step 1 :    At first, the objective function $O(s)$ is constructed with the set of $s = (s_1, s_2 \ldots s_d)^T$;
Step 2 :    Initialize the set of populations of k number of spider monkeys $s_i$ with $1 \le i \le k$, and its switching probability $\alpha_p \in$ [0, 1] with the maximum number of iterations;
Step 3 :    While $(l < Max_{itr})$ do.
Randomly select the spider monkeys for computing the fitness function by using Equations (5)–(7);
Verify the value of $M_i = O\left(s_i^{l+1}\right)$ for computing the fitness value;
While the fitness of $s_i$ is not at $(l < Itr_{max})$ do
Split the entire set of population $s_i$ with $1 \le i \le n$ into g number of groups;
//Local and global leader phase
Update the position of monkeys and global leader as shown in Equations (8)–(10);
//Learning phase
Select the best global leader based on the probability as defined in Equation (11);
Update the position of global & local leaders, and compute the fitness value for the leaders;
Group members can update their position by using Equation (12);
$Itr = Itr + 1$;
    End;
Step 4 :    If $(M_i > M_j)$ then
$M_j \leftarrow M_i$; //Replace the old solution with the new solution;
End if;
Step 5 :    If $(rand$ [0, 1] $< \alpha_p)$ then
Re-initialize the entire population with the group members;
Obtain the global best solution;
End if;
Step 6 : If $(M_i < M_{min})$ //Old solution is replaced with the new solution
    $Opt_{a(i)} = s_i$;
$M_i = M_{min}$; //Arrange the most feasible solutions for determining the current best solution;Increment the count l by 1;
Return the best optimal solution as $Opt_{a(i)}$;
    End;
```

3.3. Deep Sequential Long Short Term Memory (DS-LSTM) Classification Model

In this stage, the selected optimal number of features have been utilized by the classifier for training the model. Here, the deep sequential long short term memory (DS-LSTM) mechanism is employed to identify the intrusions from the SCADA dataset, based on the optimal number of features. It is a kind of machine learning classification mechanism and is more suitable for solving the complex prediction problems. The hyper-parameters play a vital role in the deep learning classification techniques, because they have a great impact on determining the performance of a classifier. In the existing works, the hyper-parameter tuning is performed in the deep learning models based on the random and grid search, but it is not more efficient. Hence, the proposed work aims to utilize an optimization technique for tuning the hyper-parameters. Typically, optimizing the hyper-parameters is one of the crucial processes, so it is required for deep understanding of the underlying model. Hence, the proposed work utilizes an optimization model for optimizing the hyper-parameters of a classifier, which helps to obtain improved performance results. Then, the RMSprop optimizer has been used to optimize the value of the hyper-parameters, which helps to obtain an increased training and testing accuracy. Here, the main purpose of optimizing the hyper-parameters is to increase the training and testing accuracy of classifier. In the proposed system, the different types of hyper-parameters used in the classification are as follows: learning rate, number of epochs, hidden layers, and batch size. The primary advantages of using this technique are reduced time consumption for training and testing, increased accuracy, detection rate, and minimized misclassification rate. In the proposed system, the parameter tuning process [40] has been performed by using the optimization technique that helps to efficiently improve the detection rate of proposed IDS. During this process, the optimal set of features, learning model, and label are taken as the inputs, and the predicted label is produced as the classified output. Initially, the deterministic rules $\Delta D_r(x)$ are computed according to the logical vector σ and featured data $Opt_{a(i)}$, as shown below:

$$\Delta D_r(x) = k'_v(net_v(x)(\tau - Opt_{ai}(x))) \tag{13}$$

After that, the feature map has been extracted by applying the convolutional operation across two set of data as shown below:

$$c^v = Opt_{ai}(x) + (\Delta D_r(x) + Opt_{ai}(x)) \tag{14}$$

Based on the value of target vector, the trail vector is computed by using the following model:

$$Ta^v_{i,j} = \begin{cases} c^v_{i,j}\ if\ Cla_L == 1 \\ \Delta D^v_{i,j}\ else \end{cases} \tag{15}$$

where, $Ta^v_{i,j}$ indicates the trail vector, Cla_L is the classified label, and $c^v_{i,j}$ is the convolutional vector. According to the weight value, the dropout factor is estimated for the v-th target vector, in which the neurons are randomly selected with respect to the specialization function as shown below:

$$T_U(x) = \frac{1}{2}\left(x - \sum_{x=1}^{n} \partial\, \omega'_x\ Ta^v\right) \tag{16}$$

where, $T_U(x)$ is the training data, ∂ indicates the dropout factor, ${\omega_x}'$ denotes the weight value, and Ta^v is the target vector. Consequently, the memory cells are updated with the forward pass as shown below:

$${m_c}^x = T_U(x) \odot g^x + k'_v(net_v(x)) \odot\ {m_c}^x \tag{17}$$

where, m_c is the memory cells, and g^x comprises both the feature map and feedback. Subsequently, the obtained feature values are passed to the sigmoid layer of the LSTM, where the distributed probability is estimated for each class as shown below:

$$DiP_{sd}(C_O) = \frac{e^{C_U^d}}{1 + e^{C_U^d}} \tag{18}$$

where, DiP_{sd} is the distributed probability of sigmoid function, C_O denotes the output class, and C_U^d indicates the output value with d-th class. Then, the binary cross entropy is estimated for analyzing the disparity across the definite segments that are used to attain the probability distribution function as shown below:

$$P_l = \sum_{i=1}^{v} D\left(C_U^i\right) a \log\left(DiP_{sd}\left(C_U^d\right)\right) \tag{19}$$

At last, the output predicted label is obtained as follows:

$$C_O = C_U(DiP_{sd} \leq 1) \tag{20}$$

Then, the RMSprop optimizer has been used to optimize the value of the hyper-parameters, (as showing in Algorithm 2) which helps to obtain an increased training and testing accuracy.

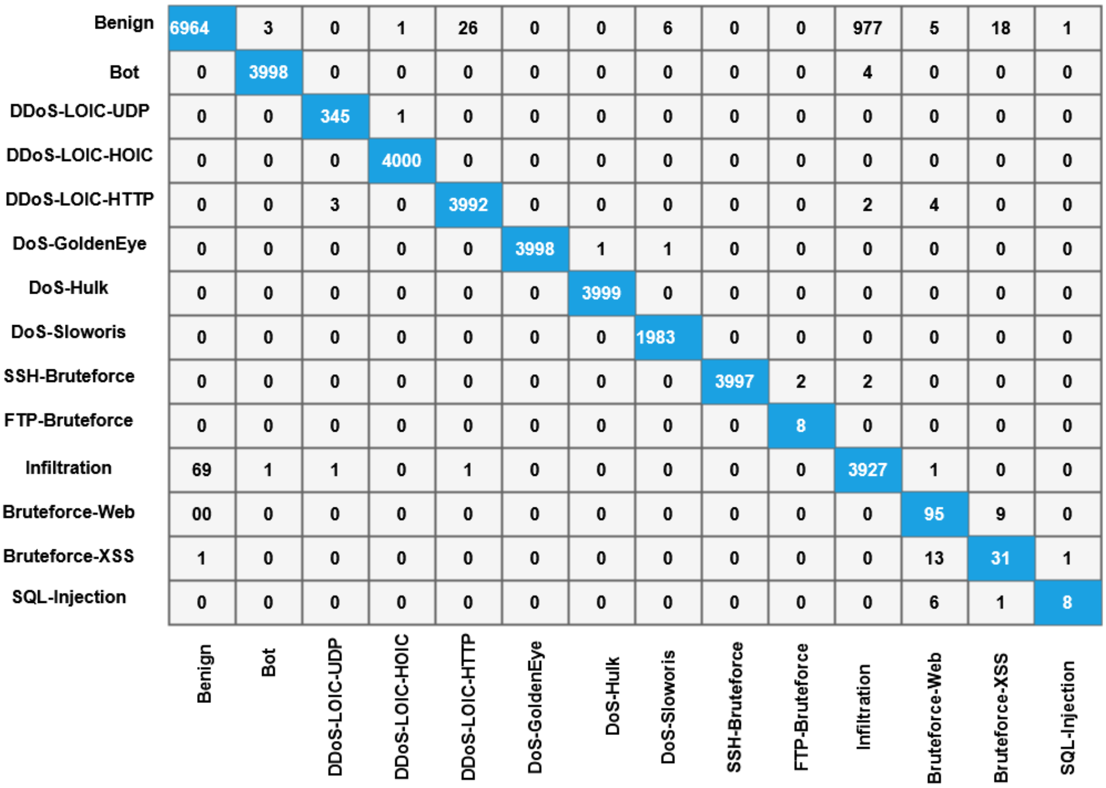
Algorithm 2 Deep Sequential Long Short Term Memory (DS-LSTM) Classification

```
Input: Optimal set of features $Opt_{a(i)}$, learning model, and Label $C_U$;
Output: Classified label $C_O$;
Step 1: Compute the deterministic rules $\Delta D_r(x)$ with respect to the logical vector $\sigma$ and featured data $Opt_{a(i)}$ by using Equation (13);
Step 2: Estimate the feature map based on the convolutional operation as shown in Equation (14);
Step 3: Compute the trail vector according to the target vector by using Equation (15);
Step 4 : Based on the obtained target vector and weight value, the dropout factor $\partial$ is estimated as shown in Equation (16);
Step 5 : Consequently, the memory cells $m_c$ are updated with the feature map and feedback value as represented in Equation (17);
Step 6 : The distributed probability $DiP_{sd}$ function is computed for each class of data by using Equation (18);
Step 7: Compute the binary cross entropy for the definite segments as shown in Equation (19);
Step 8 : Finally, the output classified label $C_O$ is predicted as represented in Equation (20);
```

4. Results and Discussions

This section evaluates the results of the proposed GDSMO-DSLSTM intrusion detection system using various performance measures. First, the different types of SCADA IDS datasets such as CSE-CIC-IDS 2018, NSL-KDD, BoT-IoT, and ICS network traffic datasets have been considered to validate this scheme's performance. Then, the results of both conventional and proposed intrusion detection methodologies are validated and compared by using various performance measures such as accuracy, precision, F1-score, true positive rate (TPR), false positive rate (FPR), detection rate, and false acceptance rate (FAR). Table 1 shows the attacking details of the CSE-CIC-IDS 2018 dataset, which comprises the different types of attacks related to bot, DDoS, DoS, brute force, and injection. Then, its corresponding confusion matrix and ROC analysis have been evaluated by using the proposed GDSMO-DSLSTM system, as shown in Figures 2 and 3, respectively. Similarly, the dataset description with the attacking details, confusion matrix, and ROC analysis for the BoT-IoT dataset is presented in Table 2, Figures 4 and 5, correspondingly. Then, the NSL-KDD dataset is also described with its features, confusion matrix, and ROC in

Figures 6–8. These evaluations show that the proposed intrusion detection system could efficiently predict the attacks of the given datasets with increased TPR.



	Benign	Bot	DDoS-LOIC-UDP	DDoS-LOIC-HOIC	DDoS-LOIC-HTTP	DoS-GoldenEye	DoS-Hulk	DoS-Sloworis	SSH-Bruteforce	FTP-Bruteforce	Infiltration	Bruteforce-Web	Bruteforce-XSS	SQL-Injection
Benign	6964	3	0	1	26	0	0	6	0	0	977	5	18	1
Bot	0	3998	0	0	0	0	0	0	0	0	4	0	0	0
DDoS-LOIC-UDP	0	0	345	1	0	0	0	0	0	0	0	0	0	0
DDoS-LOIC-HOIC	0	0	0	4000	0	0	0	0	0	0	0	0	0	0
DDoS-LOIC-HTTP	0	0	3	0	3992	0	0	0	0	0	2	4	0	0
DoS-GoldenEye	0	0	0	0	0	3998	1	1	0	0	0	0	0	0
DoS-Hulk	0	0	0	0	0	0	3999	0	0	0	0	0	0	0
DoS-Sloworis	0	0	0	0	0	0	0	1983	0	0	0	0	0	0
SSH-Bruteforce	0	0	0	0	0	0	0	0	3997	2	2	0	0	0
FTP-Bruteforce	0	0	0	0	0	0	0	0	0	8	0	0	0	0
Infiltration	69	1	1	0	1	0	0	0	0	0	3927	1	0	0
Bruteforce-Web	00	0	0	0	0	0	0	0	0	0	0	95	9	0
Bruteforce-XSS	1	0	0	0	0	0	0	0	0	0	0	13	31	1
SQL-Injection	0	0	0	0	0	0	0	0	0	0	0	6	1	8

Figure 2. Confusion matrix for the CSE-CIC-IDS 2018 dataset.

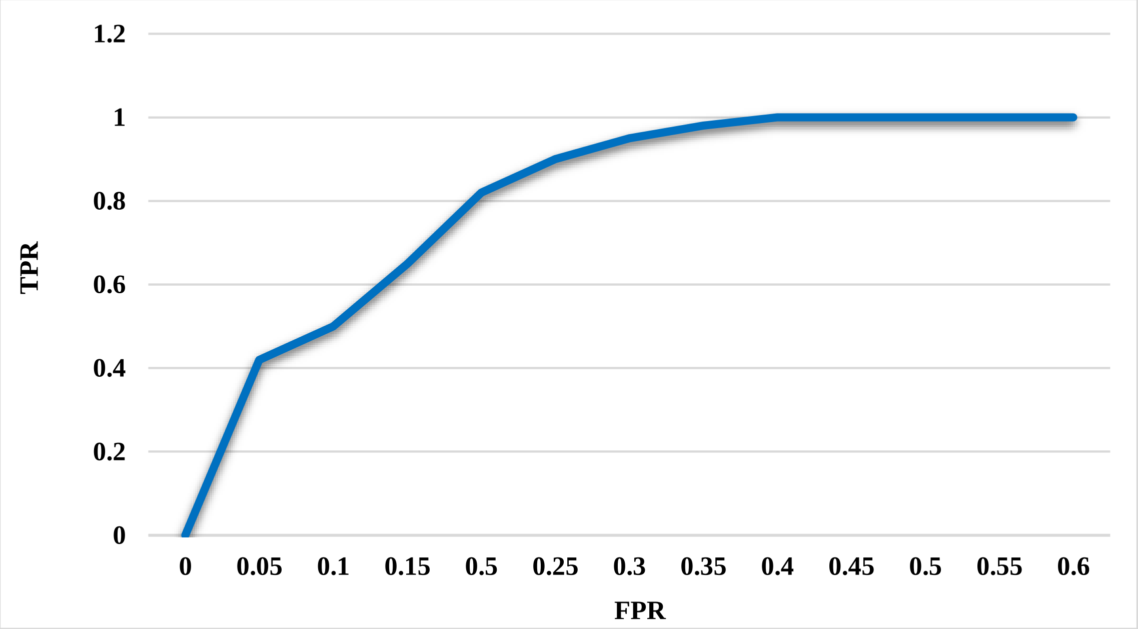


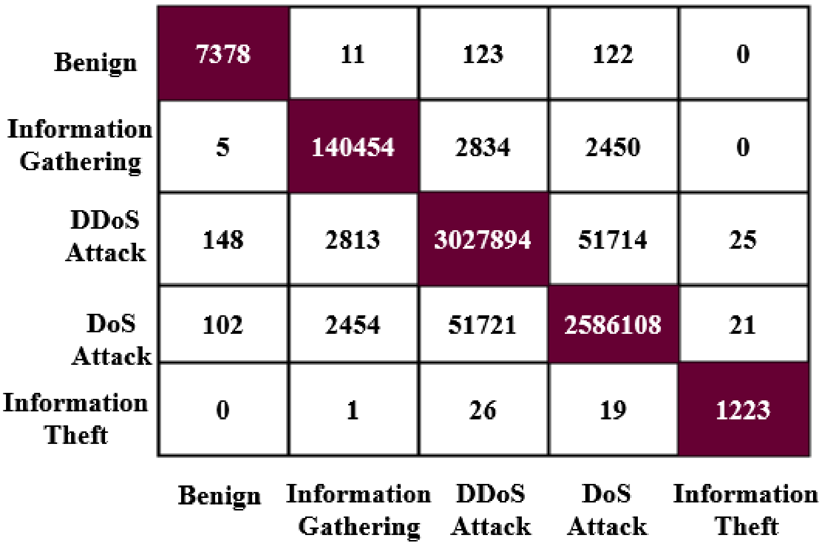
Figure 3. ROC analysis for the CSE-CIC-IDS 2018 dataset.

Table 1. CSE-CIC-IDS 2018 dataset.

Attack Types	Size
Benign	736,521
Bot	143,010
DDoS-LOIC-UDP	7085
DDoS-LOIC-HOIC	1,082,293
DDoS-LOIC-HTTP	296,084
DoS-GoldenEye	30,585
DoS-Hulk	90,051
DoS-Sloworis	13,475
SSH-Bruteforce	94,237
FTP-Bruteforce	193,360
Infiltration	209
Bruteforce-Web	268
Bruteforce-XSS	117
SQL-Injection	53

Table 2. BoT-IoT dataset.

Category	Type of Attack	Flow Count
Benign	Benign	9543
Information gathering	Service scanning	1,463,364
	OS Fingerprinting	358,275
DDoS attack	DDoS TCP	19,547,603
	DDoS UDP	18,965,106
	DDoS HTTP	19,771
DoS attack	DoS TCP	12,315,997
	DoS UDP	20,659,491
	DoS HTTP	29,706
Information theft	Key logging	1469
	Data theft	118
Total		73,370,443



	Benign	Information Gathering	DDoS Attack	DoS Attack	Information Theft
Benign	7378	11	123	122	0
Information Gathering	5	140454	2834	2450	0
DDoS Attack	148	2813	3027894	51714	25
DoS Attack	102	2454	51721	2586108	21
Information Theft	0	1	26	19	1223

Figure 4. Confusion matrix for the BoT-IoT dataset.

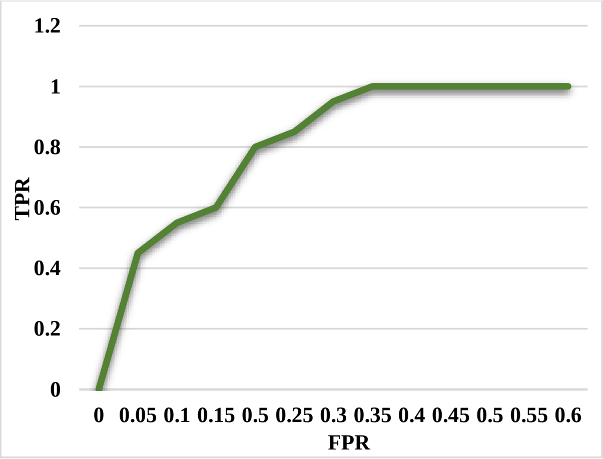
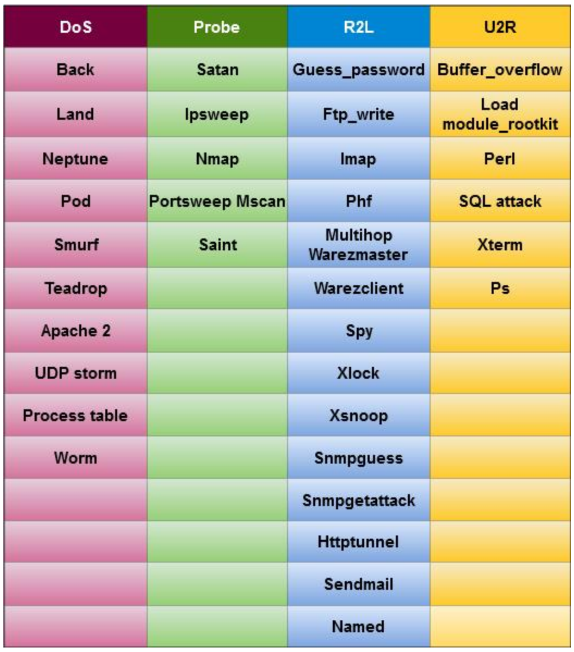
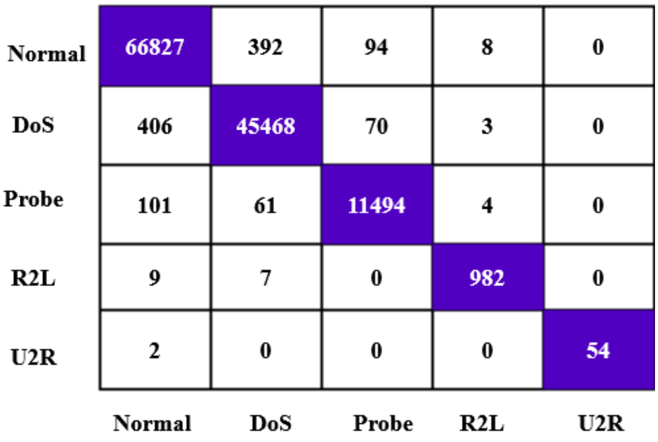


Figure 5. ROC analysis for the BoT-IoT dataset.



DoS	Probe	R2L	U2R
Back	Satan	Guess_password	Buffer_overflow
Land	Ipsweep	Ftp_write	Load module_rootkit
Neptune	Nmap	Imap	Perl
Pod	Portsweep Mscan	Phf	SQL attack
Smurf	Saint	Multihop Warezmaster	Xterm
Teadrop		Warezclient	Ps
Apache 2		Spy	
UDP storm		Xlock	
Process table		Xsnoop	
Worm		Snmpguess	
		Snmpgetattack	
		Httptunnel	
		Sendmail	
		Named	

Figure 6. NSL-KDD dataset.



	Normal	DoS	Probe	R2L	U2R
Normal	66827	392	94	8	0
DoS	406	45468	70	3	0
Probe	101	61	11494	4	0
R2L	9	7	0	982	0
U2R	2	0	0	0	54

Figure 7. Confusion matrix for the NSL-KDD dataset.

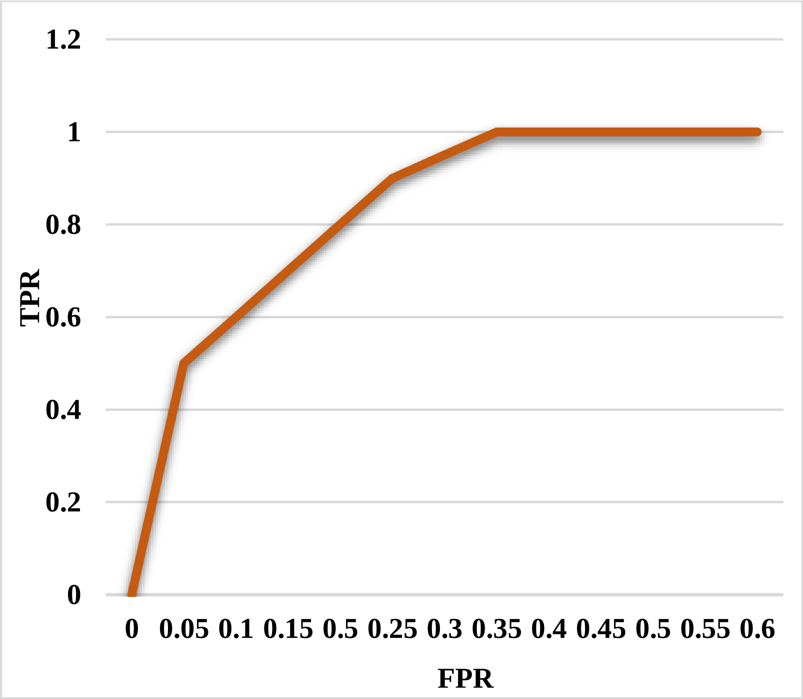


Figure 8. ROC analysis for the NSL-KDD dataset.

4.1. Simulation Analysis

For validating the performance of the proposed security mechanism, various measures such as accuracy, FPR, TPR, F1-score, and recall are computed, and the results were obtained by using the MATLAB simulation tool. Figure 9 shows the accuracy and TPR of the proposed optimization-based classification methodology concerning various iterations. Similar to that, Figure 10 estimates the F1-score and FPR of the proposed mechanism for the different number of operations. Figure 11a,b show the proposed mechanism's TPR, FPR, accuracy, and F1-score under varying iterations. According to these evaluations, it is analyzed that the proposed technique provides increased accuracy, F1-score, TPR, and reduced FPR values with a reduced number of operations. Consequently, the overall performance of the proposed system is validated and tested for the given datasets, as shown in Figure 12.

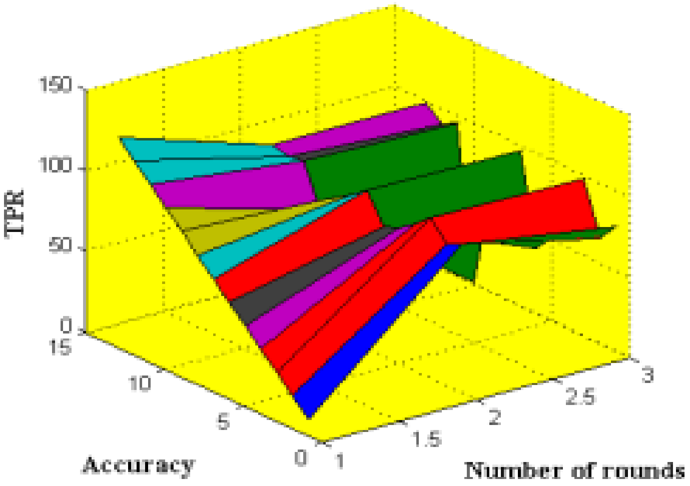


Figure 9. Accuracy vs. number of rounds.

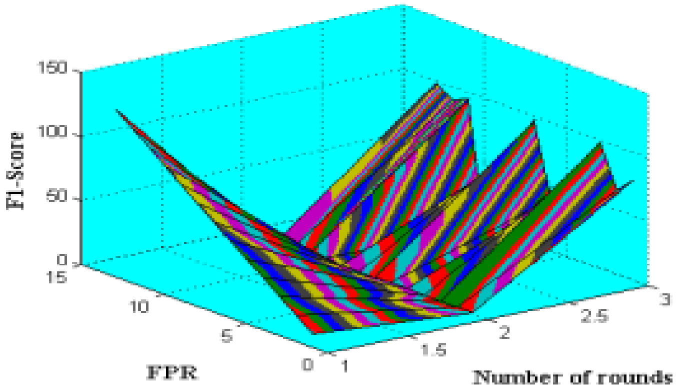


Figure 10. F1-score vs. number of rounds.

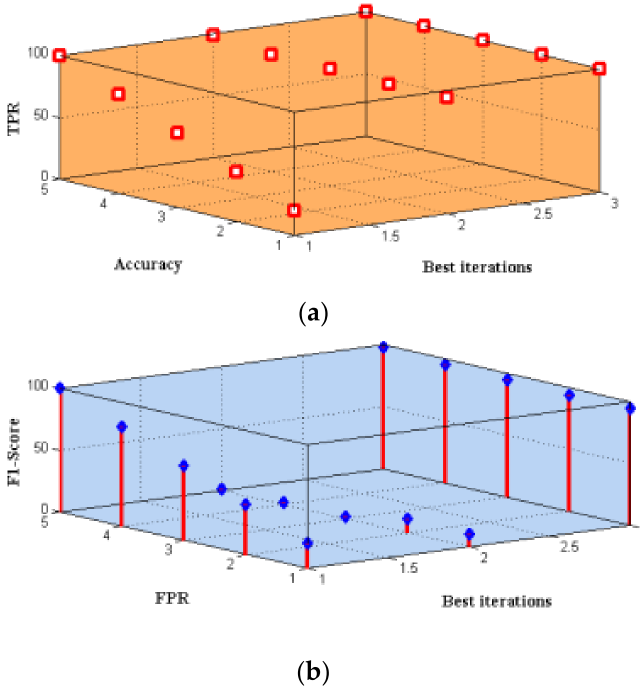


Figure 11. (**a**). Accuracy with respect to best iterations and (**b**). F1-score with respect to best iterations.

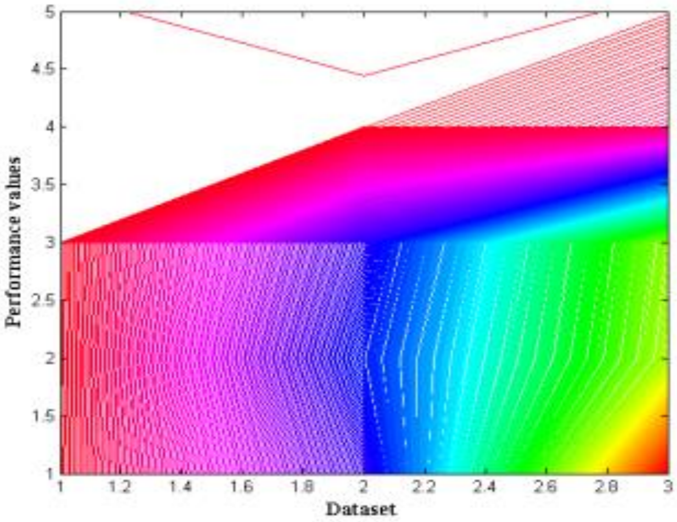


Figure 12. Overall performance analysis.

Then, the FAR and detection rate of the proposed techniques are validated for the different types of datasets, as depicted in Figures 13 and 14, respectively. To assess the improved performance rate of the proposed classification technique using F1-score, recall and accuracy measures are shown in Figure 15. The obtained results state that the proposed technique provides improved performance results for the all the IDS datasets.

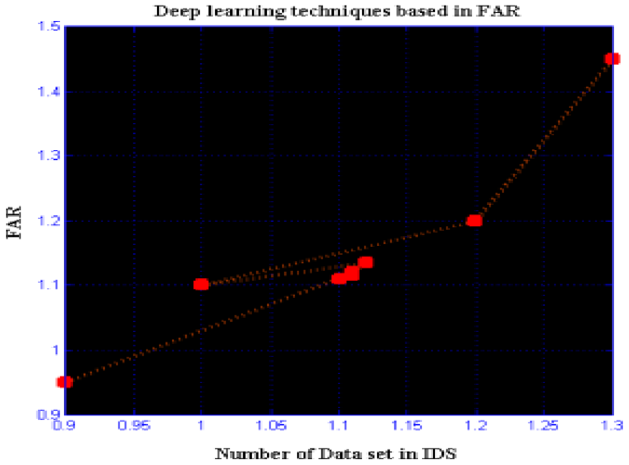


Figure 13. Analysis of FAR.

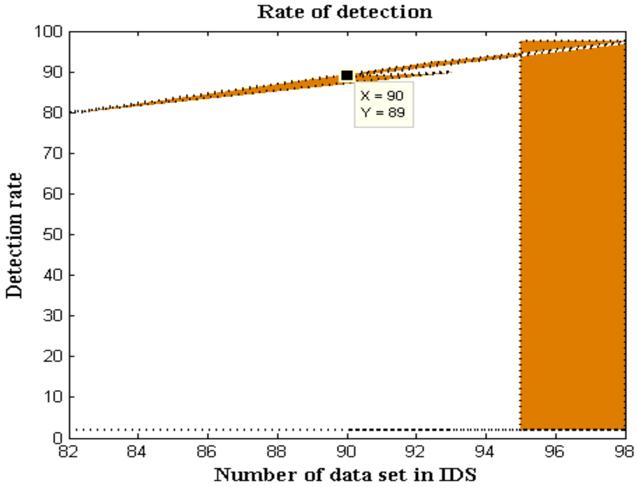


Figure 14. Detection rate.

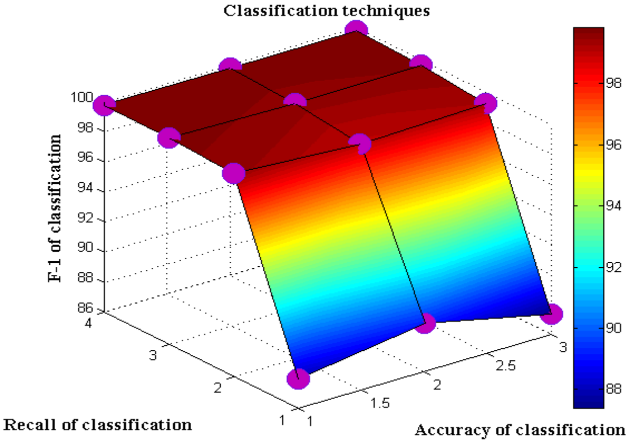


Figure 15. Accuracy, recall, and F1-score analysis.

4.2. Comparative Analysis

Table 3 and Figure 16 compare the conventional [41] and proposed intrusion detection and classification methodologies for the CSE-CIC-IDS 2018 dataset, based on the measures of accuracy, TPR, FPR, and F1-score. Typically, the efficiency of any detection and classification system is evaluated using these measures. Additionally, the overall performance of the IDS approach significantly depends on the accuracy of detection. Therefore, the accuracy, TPR, FPR and F1-score have been increasingly used to validate security systems' detection efficiency. These measures are computed by using the following models:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

$$\text{Recall or TPR} = \frac{TP}{TP + FN} \tag{23}$$

$$\text{F1} - \text{score} = \frac{2TP}{2TP + FP + FN} \tag{24}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{25}$$

where, *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative. The evaluation shows that the proposed GDSMO-DSLSTM outperforms the other techniques with increased accuracy, TPR, F1-score, and reduced FPR, because the clustering-based optimization and classification processes help obtain an improved performance during the detection of intrusions from the datasets. Table 4 and Figure 17 validate and compare the existing and proposed machine learning-based classification techniques used to detect intrusions in the SCADA systems based on accuracy, TPR, FPR, and F1-score. The obtained results also depicted that the proposed GDSMO-DSLSTM technique improves performance value over the other methods. This is because the clustering and optimal parameter tuning help to precisely locate the intrusions from the datasets based on the global fitness value. Moreover, the performance of detection depends on the quality of the input dataset, hence, the attribute normalization helps to increase the quality of data. Specifically, the multifacet clustering splits the preprocessed into a group of chunks, which is more helpful to process the dataset for classification.

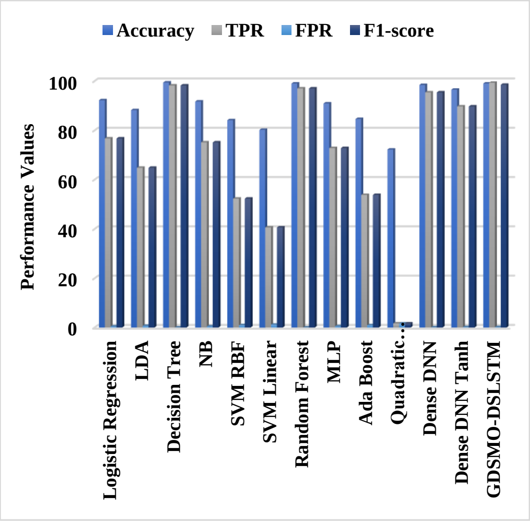


Figure 16. Performance analysis of existing and proposed classification approaches using the CSE-CIC-IDS 2018 dataset.

Table 3. Comparative analysis between existing and proposed mechanisms using the CSE-CIC-IDS 2018 dataset.

Methods	Accuracy	TPR	FPR	F1-Score
Logistic Regression	92.2	76.7	0.46	76.7
LDA	88.2	64.8	0.70	64.8
Decision Tree	99.4	98.2	0.03	98.2
NB	91.7	75.1	0.49	75.1
SVM RBF	84.1	52.3	0.95	52.3
SVM Linear	80.2	40.6	1.18	40.6
Random Forest	99	97	0.05	97
MLP	90.9	72.8	0.54	72.8
Ada Boost	84.6	53.8	0.92	53.8
Quadratic Discriminant Analysis	72.2	1.66	1.66	1.66
Dense DNN	98.4	95.4	0.09	95.4
Dense DNN Tanh	96.5	89.7	0.20	89.7
Proposed GDSMO-DSLSTM	99	99.3	0.18	98.5

Table 4. Analysis based on accuracy, TPR, FPR, and F1-score.

Methods	Accuracy	TPR	FPR	F1-Score
ABOD	94.4	100	10.1	94.2
Isolation Forest	93.8	99.9	11.1	93.7
LOF	94.4	100	1.01	94.2
Auto Encoder	95.14	100	0.96	95.33
GDSMO-DSLSTM	98.8	100	0.85	98

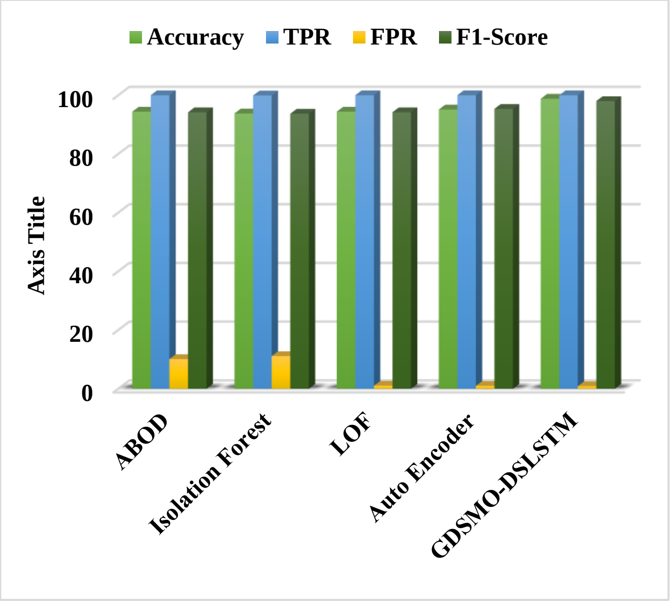


Figure 17. Comparison between existing and proposed classification techniques based on the measures of accuracy, TPR, FPR, and F1-score.

Table 5 and Figure 18 compare the conventional [42] proposed intrusion detection and classification techniques based on accuracy, detection rate, and F1-score, where the SCADA network dataset has been utilized to assess the results. Typically, the detection rate and accuracy are the essential parameters used for validating the proficiency and concert of security systems. Here, the detection rate is used to determine how accurately the IDS can identify the attacks from the datasets with increased speed and reduced time consumption. Based on the evaluations, it is perceived that the proposed GDSMO-DSLSTM technique provides increased accuracy, detection rate, and F1-score values compared to the other methods, which shows the overall improved performance rate of the proposed system.

Table 5. Accuracy, detection rate, and F1-score of existing and proposed classification techniques using the SCADA network dataset.

Techniques	Accuracy	Detection Rate	F1-Score
Decision Forest	99.72	94.12	80.26
Boosted Decision Forest	99.77	93.14	84.67
Decision Jungle	99.79	93.97	85.08
Cyber physical model	99.79	99.78	98.7
Proposed GDSMO-DSLSTM	99.8	99.85	99.8

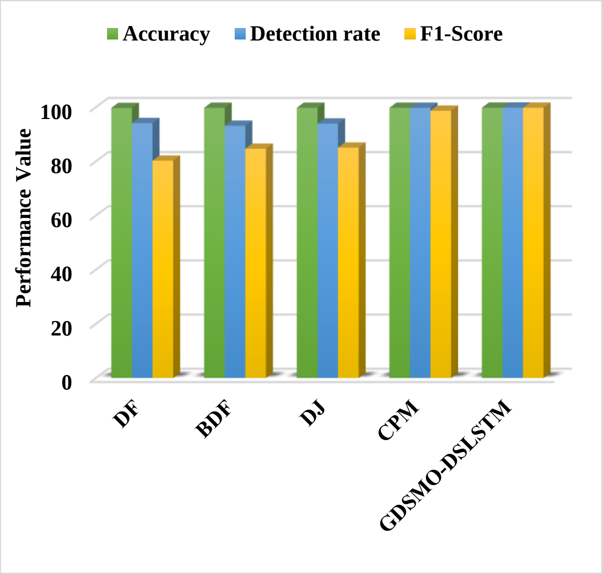


Figure 18. Comparative analysis between existing and proposed techniques using the SCADA network dataset.

Table 6 and Figure 19 compare the existing [43] and proposed deep learning techniques used to develop the IDS frameworks, based on the false acceptance rate (FAR) measure. Both datasets, such as CSE-CIC-IDS 2018 and BoT-IoT, have been taken for validation and comparison. Similarly, the detection rate of existing and proposed deep learning models are compared using these datasets, as shown in Table 7 and Figure 20. According to these evaluations, it is observed that the proposed GDSMO-DSLSTM provides a reduced FAR and increased detection rate for both datasets, when compared to the other techniques. This is because the proposed optimization technique supports the training the deep learning classifier with the best optimal features, which avoids an increased FAR of classification.

Table 6. Comparative analysis between the existing and proposed deep learning techniques based on FAR.

Techniques	CSE-CIC-IDS 2018	BoT-IoT
DNN	1.3	1.45
RNN	1.2	1.2
CNN	1	1.1
RBM	1.12	1.135
DBN	1.11	1.12
DBM	1.11	1.115
DA	1.10	1.11
GDSMO-DSLSTM	0.9	0.95

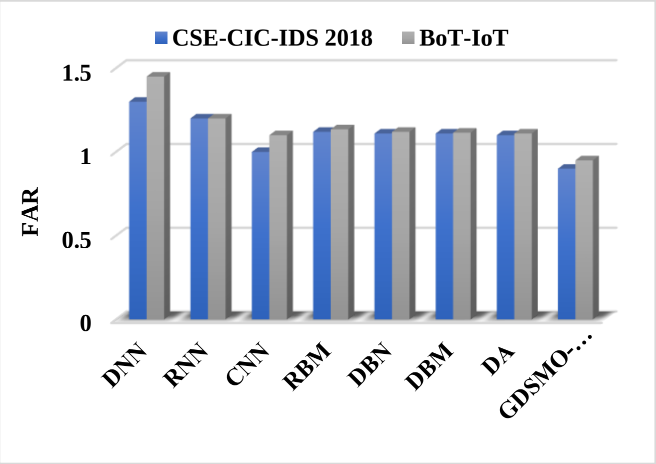


Figure 19. FAR of existing and proposed deep learning mechanisms for both the CSE-CIC-IDS 2018 and BoT-IoT datasets.

Table 7. Comparative analysis between the existing and proposed deep learning techniques based on detection rate.

Techniques	CSE-CIC-IDS 2018	BoT-IoT
DNN	95	97.5
RNN	98	97.5
CNN	98	97.5
RF	92.5	92.5
NB	82	80
SVM	93	90
ANN	90	89
GDSMO-DSLSTM	98	97

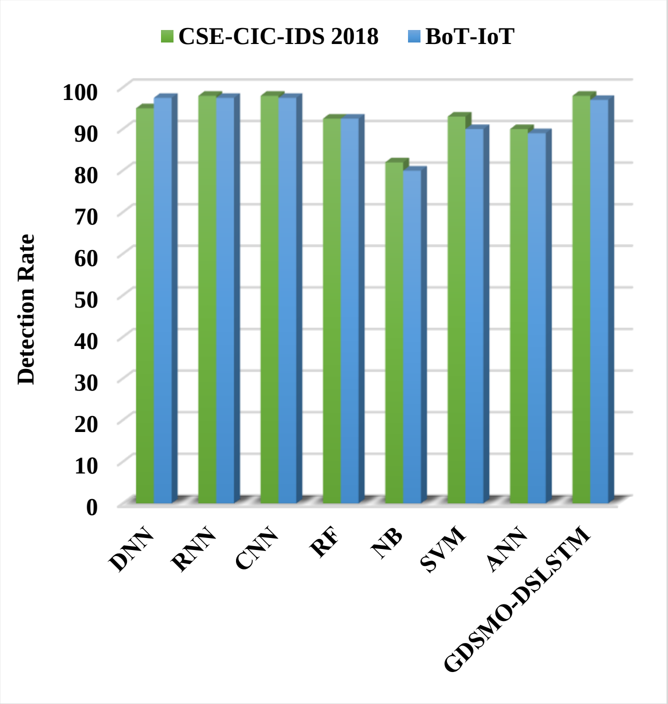


Figure 20. Detection rate of existing and proposed deep learning mechanisms for both the CSE-CIC-IDS 2018 and BoT-IoT datasets.

Table 8 and Figure 21 compare the precision, recall, and f1-measure of both existing [27] and proposed classification techniques using the omni-attacks dataset. The precision and recall measures are generally used in all classification and detection application systems to assess the classifier's performance and efficiency. Based on these results, it is evident that the proposed GDSMO-DSLSTM technique provides increased precision, recall, and f1-measure values compared to the other methods. Furthermore, the optimal parameter tuning attains improved performance outcomes over the different classifiers.

Table 8. Precision, recall and f1-measure of existing and proposed classification techniques.

Methods	Precision	Recall	F1-Measure
FNN	88	89.2	87.4
LSTM	99.54	99.01	99.27
Ensemble Learning	99.76	99.57	99.68
GDSMO-DSLSTM	99.8	99.8	99.85

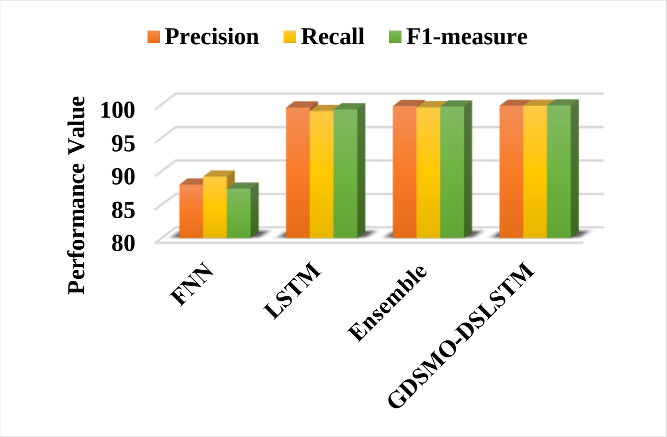


Figure 21. Comparative analysis based on precision, recall, and f1-measure.

5. Conclusions

This paper presents a classy multifacet clustering-based optimization and classification methodology for detecting intrusions from the SCADA systems. The main contribution of this work is to develop an intelligent IDS framework by using the fusion of methods for obtaining an increased detection accuracy, reduced false positives, error rate, and complexity. The most popular IDS datasets have been utilized to implement and validate the proposed security system. The dataset normalization and preprocessing operations have been performed to eliminate irrelevant attributes and balance the data. Consequently, the MDCM technique is applied to group the attributes into the form of clusters based on the distance value. The main purpose of implementing the clustering technique is to simplify the process of intrusion detection and classification with an increased speed of processing. Then, the GDSMO technique is employed to optimally select the best features for training the classifier model, which helps reduce the time taken for dataset training and testing. The switching probability, weight value, and fitness value have been computed during this process for selecting the optimal parameters to improve the classification.

Moreover, the DS-LSTM-based deep learning classifier is deployed for spotting the intrusions from the clustered datasets based on the optimal set of features. The primary advantages of using this technique are reduced time consumption for training and testing, increased accuracy and detection rate, and minimized misclassification rate. Finally, the performance of the proposed GDSMO-DSLSTM-based IDS is validated and compared with the recent state-of-the-art models by using the measures of accuracy, precision, recall, F1-score, FAR, and detection rate. The evaluation states that the proposed GDSMO-DSLSTM technique outperforms the other approaches with improved performance values.

In future, the proposed work can be enhanced by developing a secured communication medium for protecting the SCADA systems from internal and external threats. Additionally, the major properties such as integrity, scalability, intrusion tolerance, and self-healing can be satisfied by designing an effectively secured SCADA architecture.

Author Contributions: Data curation: K.H.A. and A.Y.; Writing original draft: H.M.; Supervision: S.S., A.O.K. (Alaa O. Khadidos) and A.O.K. (Adil O. Khadidos); Project administration: S.S., A.O.K. (Alaa O. Khadidos) and A.O.K. (Adil O. Khadidos); Conceptualization: H.M. and S.S.; Methodology: S.S. and H.M.; Validation: K.H.A. and A.Y.; Visualisation: H.M. and S.S.; Resources: A.O.K. (Alaa O. Khadidos) and A.O.K. (Adil O. Khadidos); Review—Editing: S.S., A.O.K. (Alaa O. Khadidos) and A.O.K. (Alaa O. Khadidos); Funding acquisition: A.O.K. (Alaa O. Khadidos), A.O.K. (Adil O. Khadidos), K.H.A. and A.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia has funded this project, under grant no. (RG-4-611-43). The authors, therefore, acknowledge with thanks to DSR for technical and financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Upadhyay, D.; Sampalli, S. SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Comput. Secur.* **2020**, *89*, 101666. [CrossRef]
2. Yadav, G.; Paul, K. Architecture and security of SCADA systems: A review. *Int. J. Crit. Infrastruct. Prot.* **2021**, *34*, 100433. [CrossRef]
3. Pliatsios, D.; Sarigiannidis, P.; Lagkas, T.; Sarigiannidis, A.G. A survey on SCADA systems: Secure protocols, incidents, threats and tactics. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1942–1976. [CrossRef]
4. Suaboot, J.; Fahad, A.; Tari, Z.; Grundy, J.; Mahmood, A.N.; Almalawi, A.; Zomaya, A.Y.; Drira, K. A taxonomy of supervised learning for idss in scada environments. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–37. [CrossRef]
5. el Kalam, A.A. Securing SCADA and critical industrial systems: From needs to security mechanisms. *Int. J. Crit. Infrastruct. Prot.* **2021**, *32*, 100394. [CrossRef]
6. Rakas, S.V.B.; Stojanović, M.D.; Marković-Petrović, J.D. A review of research work on network-based scada intrusion detection systems. *IEEE Access* **2020**, *8*, 93083–93108. [CrossRef]
7. Alimi, O.A.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, K.O.A. A review of research works on supervised learning algorithms for SCADA intrusion detection and classification. *Sustainability* **2021**, *13*, 9597. [CrossRef]
8. Al-Asiri, M.; El-Alfy, E.-S.M. On using physical based intrusion detection in SCADA systems. *Procedia Comput. Sci.* **2020**, *170*, 34–42. [CrossRef]
9. Qian, X.; Du, B.; Chen, B.; Qu, K.; Zeng, K.; Liu, J. Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry. *IEEE Access* **2020**, *8*, 147471–147481. [CrossRef]
10. Maglaras, L.; Cruz, T.; Ferrag, M.A.; Janicke, H. Teaching the process of building an Intrusion Detection System using data from a small-scale SCADA testbed. *Internet Technol. Lett.* **2020**, *3*, e132. [CrossRef]
11. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2559–2574. [CrossRef]
12. Qassim, Q.S.; Jamil, N.; Mahdi, M.N.; Rahim, A.A.A. Towards scada threat intelligence based on intrusion detection systems—A short review. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU), Selangor, Malaysia, 24–26 August 2020; pp. 144–149. [CrossRef]
13. Chaithanya, P.; Priyanga, S.; Pravinraj, S.; Sriram, V.S. SSO-IF: An Outlier Detection Approach for Intrusion Detection in SCADA Systems. In *Inventive Communication and Computational Technologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 921–929.
14. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 1104–1116. [CrossRef]
15. Shitharth, S.; Sangeetha, K.; Kumar, B.P. Integrated probabilistic relevancy classification (prc) scheme for intrusion detection in scada network. In *Design Frameworks for Wireless Networks*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 41–63.
16. Gaiceanu, M.; Stanculescu, M.; Andrei, P.C.; Solcanu, V.; Gaiceanu, T.; Andrei, H. Intrusion Detection on ICS and SCADA Networks. In *Recent Developments on Industrial Control Systems Resilience*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 197–262.

17. Sangeetha, K.; Shitharth, S.; Mohammed, G.B. Enhanced SCADA IDS Security by Using MSOM Hybrid Unsupervised Algorithm. *Int. J. Web-Based Learn. Teach. Technol. (IJWLTT)* **2022**, *17*, 1–9. [CrossRef]
18. Rajesh, L.; Satyanarayana, P. Evaluation of Machine Learning Algorithms for Detection of Malicious Traffic in SCADA Network. *J. Electr. Eng. Technol.* **2021**, 1–16. [CrossRef]
19. Yin, X.C.; Liu, Z.G.; Nkenyereye, L.; Ndibanje, B. Toward an applied cyber security solution in IoT-based smart grids: An intrusion detection system approach. *Sensors* **2019**, *19*, 4952. [CrossRef]
20. Yang, H.; Cheng, L.; Chuah, M.C. Deep-learning-based network intrusion detection for SCADA systems. In Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 1–7.
21. Ozkan-Okay, M.; Samet, R.; Aslan, Ö.; Gupta, D. A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access* **2021**, *9*, 157727–157760. [CrossRef]
22. Khan, I.A.; Pi, D.; Khan, Z.U.; Hussain, Y.; Nawaz, A. HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. *IEEE Access* **2019**, *7*, 89507–89521. [CrossRef]
23. Singh, V.K.; Ebrahem, H.; Govindarasu, M. Security evaluation of two intrusion detection systems in smart grid scada environment. In Proceedings of the 2018 North American Power Symposium (NAPS), Fargo, ND, USA, 9–11 September 2018; pp. 1–6.
24. Gao, J.; Gan, L.; Buschendorf, F.; Zhang, L.; Liu, H.; Li, P.; Dong, X.; Lu, T. LSTM for SCADA intrusion detection. In Proceedings of the 2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 21–23 August 2019; pp. 1–5.
25. Waagsnes, H.; Ulltveit-Moe, N. Intrusion Detection System Test Framework for SCADA Systems. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Madeira, Portugal, 22–24 January 2018; pp. 275–285.
26. Altaha, M.; Lee, J.-M.; Muhammad, A.; Hong, S. An autoencoder-based network intrusion detection system for the SCADA system. *J. Commun.* **2021**, *16*, 210–216. [CrossRef]
27. Gao, J.; Gan, L.; Buschendorf, F.; Zhang, L.; Liu, H.; Li, P.; Dong, X.; Lu, T. Omni SCADA intrusion detection using deep learning algorithms. *IEEE Internet Things J.* **2020**, *8*, 951–961. [CrossRef]
28. Tamy, S.; Belhadaoui, H.; Rabbah, M.A.; Rabbah, N.; Rifi, M. An evaluation of machine learning algorithms to detect attacks in SCADA network. In Proceedings of the 2019 7th Mediterranean Congress of Telecommunications (CMT), Fez, Morocco, 24–25 October 2019; pp. 1–5.
29. Justindhas, Y.; Jeyanthi, P. Attack detection and prevention in IoT-SCADA networks using NK-classifier. *Soft Comput.* **2022**, 1–13. [CrossRef]
30. Hopkins, S.; Kalaimannan, E. Towards establishing a security engineered SCADA framework. *J. Cyber Secur. Technol.* **2019**, *3*, 47–59. [CrossRef]
31. Teixeira, M.A.; Salman, T.; Zolanvari, M.; Jain, R.; Meskin, N.; Samaka, M. SCADA system testbed for cybersecurity research using machine learning approach. *Future Internet* **2018**, *10*, 76. [CrossRef]
32. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Efstathopoulos, G.; Karypidis, P.-A.; Sarigiannidis, A. DIDEROT: An intrusion detection and prevention system for DNP3-based SCADA systems. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual Event, Coimbra, Portugal, 25–28 August 2020; pp. 1–8.
33. Benisha, R.; Ratna, S.R. Design of intrusion detection and prevention in SCADA system for the detection of bias injection attacks. *Secur. Commun. Netw.* **2019**, *2019*, 108248. [CrossRef]
34. Li, H.; Shi, D.; Wang, W.; Liao, D.; Gadekallu, T.R.; Yu, K. Secure Routing for LEO Satellite Network Survivability. *Comput. Netw.* **2022**, 109011. [CrossRef]
35. Hariprasath, M.; Subramanian, S.; Ganesan, S.; Abirami, M. Reliable/cost-effective optimization framework for precise phasor measurement locations. *Int. J. Power Energy Syst.* **2017**, *37*. [CrossRef]
36. Yu, Y.; Rashidi, M.; Samali, B.; Mohammadi, M.; Nguyen, T.N.; Zhou, X. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. *Struct. Health Monit.* **2022**. [CrossRef]
37. Gibb, S.; La, H.M.; Louis, S. A genetic algorithm for convolutional network structure optimization for concrete crack detection. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
38. Agrawal, S.; Sarkar, S.; Alazab, M.; Maddikunta, P.K.R.; Gadekallu, T.R.; Pham, Q.-V. Genetic CFL: Hyperparameter Optimization in Clustered Federated Learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 7156420. [CrossRef]
39. Selvarajan, S.; Shaik, M.; Ameerjohn, S.; Kannan, S. Mining of intrusion attack in SCADA network using clustering and genetically seeded flora-based optimal classification algorithm. *IET Inf. Secur.* **2019**, *14*, 1–11. [CrossRef]
40. Bhattacharya, S.; Maddikunta, P.K.R.; Meenakshisundaram, I.; Gadekallu, T.R.; Sharma, S.; Alkahtani, M.; Abidi, M.H. Deep neural networks based approach for battery life prediction. *CMC-Comput. Mater. Contin.* **2021**, *69*, 2599–2615. [CrossRef]
41. Grammatikis, P.R.; Sarigiannidis, P.; Efstathopoulos, G.; Panaousis, E. ARIES: A novel multivariate intrusion detection system for smart grid. *Sensors* **2020**, *20*, 5305. [CrossRef]
42. Sheng, C.; Yao, Y.; Fu, Q.; Yang, W. A cyber-physical model for SCADA system and its intrusion detection. *Comput. Netw.* **2021**, *185*, 107677. [CrossRef]
43. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]