sustainability

MDPI

Article

Intrusion Detection Using Chaotic Poor and Rich Optimization with Deep Learning Model for Smart City Environment

Fatma S. Alrayes [1], Mashael M. Asiri [2], Mashael Maashi [3], Ahmed S. Salama [4], Manar Ahmed Hamza [5,*], Sara Saadeldeen Ibrahim [5], Abu Sarwar Zamani [5] and Mohamed Ibrahim Alsaid [5]

1 Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
2 Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia
3 Department of Software Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 103786, Riyadh 11671, Saudi Arabia
4 Department of Electrical Engineering, Faculty of Engineering & Technology, Future University in Egypt, New Cairo 11845, Egypt
5 Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj 16278, Saudi Arabia
* Correspondence: ma.hamza@psau.edu.sa

Abstract: Artificial intelligence (AI) techniques play a vital role in the evolving growth and rapid development of smart cities. To develop a smart environment, enhancements to the execution, sustainability, and security of traditional mechanisms become mandatory. Intrusion detection systems (IDS) can be considered an effective solutions to achieve security in the smart environment. This article introduces intrusion detection using chaotic poor and rich optimization with a deep learning model (IDCPRO-DLM) for ubiquitous and smart atmospheres. The IDCPRO-DLM model follows preprocessing, feature selection, and classification stages. At the initial stage, the Z-score data normalization system is exploited to scale the input data. Additionally, the IDCPRO-DLM method designs a chaotic poor and rich optimization algorithm-based feature selection (CPROA-FS) approach for selecting feature subsets. For intrusion detection, butterfly optimization algorithm (BOA) with a deep sparse autoencoder (DSAE) is used. The simulation analysis of the IDCPRO-DLM technique is studied on the benchmark CICIDS dataset and the comparison results show the better performance of the IDCPRO-DLM algorithm over recent state-of-the-art approaches with a maximum accuracy of 98.53%.

Keywords: smart cities; intrusion detection system; feature selection; deep learning; smart environment; security

check for updates

Citation: Alrayes, F.S.; Asiri, M.M.; Maashi, M.; Salama, A.S.; Hamza, M.A.; Ibrahim, S.S.; Zamani, A.S.; Alsaid, M.I. Intrusion Detection Using Chaotic Poor and Rich Optimization with Deep Learning Model for Smart City Environment. *Sustainability* **2023**, *15*, 6902. https://doi.org/10.3390/su15086902

Academic Editor: Fadi Al-Turjman

Received: 4 February 2023
Revised: 27 February 2023
Accepted: 2 March 2023
Published: 19 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Owing to the evolving advancement and fast progression in artificial technology, the ubiquitous computing method is a vital part of everyday life [1]. The common aim of the ubiquitous method is a too-quick tight combination of its coexisting ambience, human beings, and information. To devise smart ambiences, an enhancement to security, performances, and sustainability of conventional policies was essential [2]. As a community, computation resources must be treated as finite and precious, utilized if needed, and as effectually as possible. The Internet of Things (IoT) was based on interconnected smart gadgets, and different services were used to intergrade one network [3]. This allows the smart devices to collect delicate data and executes important functions, and these gadgets link and interact with each other at the highest speed and make decisions based on the indicator information. The IoT atmospheres employ cloud service as the back-end top for processing information and maintaining remote control. Client users leverage web services

or mobile applications to access information and control their gadgets. The IoT structures employ an enormous quantity of sensor nodes to extract significant data, and it can be scrutinized using the artificial intelligence (AI) method [4,5].

An IDS refers to a security methodology which majorly operates in the IoT network layer. An IDS positioned for an IoT network has the capability in the analyses data packets to produce responses realistically, examining data packets in dissimilar layers of IoT networks with dissimilar protocol stacks, and adapting to several methods in IoT systems [5,6]. An IDS can be advanced for IoT-related smart settings that need to operate in extreme circumstances of higher-volume data processing, less processing capability, and quicker response. Therefore, conventional IDS does not suit the IoT network [7]. The security of IoT is a continuous and serious issue; thus, the progression of respective mitigating techniques and understanding of security vulnerabilities of IoT mechanisms is required. Machine learning (ML) techniques are widely used in the detection of several types of assaults, and they can support the network administrator in taking corresponding initiatives for preventing intrusion [8]. However, several traditional ML approaches go to shallow learning and often highlight feature selection and engineering; it cannot proficiently solve the vast intrusion data classifying issues that appear in face of real-time network applications. Of the speedy advancement of the dataset, numerous classifying approaches may lead to diminished performance [9]. Along with that, shallow learning cannot be suitable for predicting the need for high-dimension learning with huge data and intellectual analysis. Conversely, deep learners may have a possibility to derive superior representation from the data for generating better methods. Subsequently, IDS has encountered incredible progression after dropping into an unhurried period [10].

This article presents intrusion detection using chaotic poor and rich optimization with a deep learning model (IDCPRO-DLM) for ubiquitous and smart environments. The presented IDCPRO-DLM model exploits the Z-score data normalization approach to scale the input data. In addition, the IDCPRO-DLM model designs a chaotic poor and rich optimization algorithm-related feature selection (CPROA-FS) approach for electing feature subsets. Moreover, the butterfly optimization algorithm (BOA) with a deep sparse autoencoder (DSAE) classification model is utilized to classify intrusions. The simulation analysis of the IDCPRO-DLM approach was tested employing a series of simulations.

The rest of the paper is organized as follows. Section 2 provides the related works and Section 3 offers the proposed model. Then, Section 4 presents the result analysis and Section 5 concludes the paper.

2. Related Works

Subbarayalu et al. [11] modelled the operation behaviors of IoT events with timed AC and developed a new hybrid NIDS. A web server is incorporated into IoT devices for remote accessibility, and constraint application protocol can be applied in intra and inter -smart device transmission. In [12], the authors presented various concepts for the security of wireless sensors in a smart environment. The suggested technique is dependent on electronic gadgets; the IDS minimizes the computation function of the sensor, thus increasing the efficacy. Similarly, the IDS presents the idea of a feedback signal and "trust table" applied for triggering the isolation and detection mechanisms in case of attack. Manimurugan et al. [13] projected a DL-related DBN system for IDS. Regarding anomaly detection and attacks, the CICIDS 2017 data are exploited for the assessment results of the presented method.

Efstathopoulos et al. [14] presented anomaly-based IDSs particularly developed for SG utilizing operation datasets from a real-time power plant. Especially important, various ML and DL methods have been deployed, presenting novel parameters and feature representations in a comparative analysis. Razib et al. [15] developed a DL-driven software defined networking (SDN)-assisted IDS to fight emergent cyberattacks in IoT. The suggested algorithm is able to counter a remarkable class of less frequently and commonly occurring cyberattacks in IoT communication. In [16], the authors determine exactly how

supervised ML algorithms (RF, DT, KNN, and ridge classifiers) and DRNN are employed for developing a robust IDS in the IoMT environments to forecast and classify unpredicted cyberattacks. Normalization and preprocessing of network datasets are implemented. Then, the researchers optimized the feature using a bio-inspired PSO approach.

Moradi et al. [17] designed the intelligent IDS (I-IDS) using the ML mechanism so that the attack is recognized in the IoT system. The algorithm is constructed based on the malicious and normal assaults on the information that was produced in IoT smart environments. For simulating these models, a testbed is constructed whereby a node MCU, a wireless router, and a DHT11 sensor have been employed. An adversarial or attacker system is constructed to implement sniffing and poisoning attacks via a laptop system. The node captures the sensor value and transfers the information to the ThinkSpeak framework in the normal phase through a wireless gateway in addition to the assault stage; then, the malicious attackers interpret the information, and change it whereas transferring from nodes to the ThinkSpeak server.

In [18], the authors establish a novel extraction feature and selective manners for the IDS method utilizing the benefits of SI techniques. The authors propose an extraction feature process based on CNN. Then, the authors propose an alternative selective feature (FS) system utilizing the newly established SI technique, Aquila optimizer (AQU). Fatani et al. [19] present an effective AI-based process for IDS in IoT methods. The authors present an extraction feature approach utilizing the CNNs for extracting significant features. In addition, the authors establish a novel selective feature model utilizing a novel different of the transient search optimizer (TSO) technique named TSODE, utilizing the operators of the differential evolution (DE) technique.

D'hooge et al.'s [20] purposes for demonstrating that strong common performance could not typically follow a strong classifier on the current IDS databases. Binary classification approaches in a range of algorithmic families can be trained on the attack classes of CSE-CIC-IDS2018. Talaei Khoei and Kaabouch [21] relates the efficacy of supervised and unsupervised learning approaches in detecting cybersecurity. The supervised approaches are classification and regression decision tree, gaussian naïve bayes, LR, light gradient boosting, C-SVM, and Alex neural network. The unsupervised approaches are PCA, variational autoencoder (VAE), and K-means.

3. The Proposed Model

In this article, a novel IDCRPO-DLM methodology was developed for the recognition and classification of intrusion in smart environments. The presented IDCRPO-DLM methodology incorporates Z-score normalization, CRPOA-based feature subset selection, DSAE-based intrusion classification, and BOA-based hyperparameter tuning. The overall process is revealed in Figure 1.

3.1. Data Pre-Processing

Primarily, the Z-score data normalization approach is exploited to scale the input data. It is a predictable normal and standard technique that denotes the sum of standard deviations (SD), and original data points were below or above the population mean. It possibly lies between -3 and $+3$. This normalization of the dataset to mentioned scale converts all datasets with distinct scales to the default scale.

$$Z - score = \frac{(x - \mu)}{\sigma} \quad (1)$$

where x indicates the value of a particular instance, μ represents the mean and σ denotes SD.

3.2. Process Involved in CPROA-FS Technique

As soon as the input dataset is preprocessed, the IDCPRO-DLM model developed a CPROA-FS approach to elect feature subsets. Generally, the CPROA integrates the concepts of classical PROA with the chaotic map. PROA is a new bio-inspired meta-heuristic

algorithm dependent upon the behavior of the poor and the rich. Since it suffers from low convergence speed and premature convergence, and easily traps in the local optimum, the CPROA is derived through the use of chaotic-based population initialization.

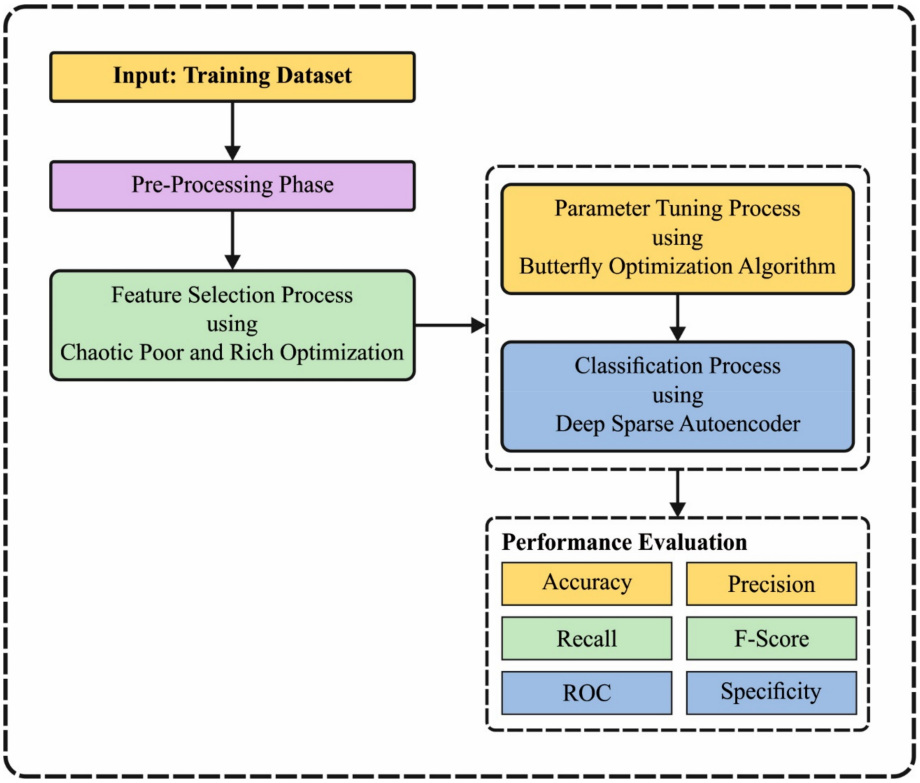


Figure 1. Block diagram of IDCPRO-DLM model.

The PROA is reliant primarily on people's wealth behaviors in society [22]. In general, the grouping of persons according to their financial condition includes the two most important groups. The initial one consist of the poor economical class persons where wealth is lower than the average. The last one is richer economic class persons where wealth exceeds the average. In this study, the solution in poor populations shifts the way of globally optimal solutions in searching regions by learning from rich solutions in the population.

Now, each individual that exists in the population is determined as a binary vector and its length corresponds to the distinct feature number that exists in the input dataset. Where $\chi = [\eta_1, \eta_2, \eta_3, \ldots, \eta_n]$, n refers to different features and every position of the solution maintains a binary value. The solution candidate encompasses poor and rich economical solutions. 'N' refers to population size. An 'N' solution is randomly produced with random numbers within zero and one. Next, the digitization method is applied to all the locations of the solution to transform the real value into binary values:

$$\chi_{i,j} = \begin{cases} 1, \ \chi_{i,j} > rand \\ 0, \ otherwise \end{cases} \tag{2}$$

In Equation (13), *rand* indicates an arbitrary number between zero and one. The solution candidate that exists in the population is sorted based on the objective function. The bottommost and topmost people in the population indicate poor and rich economic sets of persons. The major population of the PROA can be demonstrated in Equation (3).

$$POP_{Main} = POP_{rich} + POP_{poor} \tag{3}$$

People in a rich economical class shift in the way of improving the economical class gaps by observing the poor economical class. The poor people shift in the direction of decreasing the economical class by learning from the individuals in the rich class to increase their financial status. The features of rich and poor people are exploited for producing novel solutions. The motion of the rich solution is denoted by the following equation.

$$\chi^{new} = \chi^{old}_{rich,\ i,j} + \alpha * \left[\chi^{\text{old}}_{rich,i,j}\middle| - \chi^{old}_{poor,best,j}\right] \tag{4}$$

The motion of the poor solution is represented as follows.

$$\chi^{new}_{poor,i,j} = \chi^{old}_{poor,i,j} + \alpha * \left[\left(\frac{\chi^{0}_{rich,best,j} + \chi^{0}_{rich,mean,j} + \chi^{0}_{rich,worst,j}}{3}\right) - \chi^{0}_{poor,i,j}\right] \tag{5}$$

In this study, the CPROA is derived through the use of a chaotic map into the PROA to enhance the arbitrary nature (diversity) to arrive at a global minimum and escape from trapping to the local minimum which results in fast convergence. The primary point of the chaotic map is an arbitrary number in [0, 1]. The CPROA-FS approach was used to identify the optimum feature subset that enhances the classification procedure. The solution vector assessments were unchangeable and ranged from zero to one. The fitness value of the solution can be represented in a binary form. It is given below.

$$f_\theta = \alpha \cdot E + (1 - \alpha)\frac{\sum_i \theta_i}{N} \tag{6}$$

where f_θ displays FF to vector θ sized N with 1/0 units that elaborate unelected or elected feature subsets, N denotes the total quantity of features. E means the classifier error, and α defines a constant tradeoff balancing from classifier error to quantity of selected feature sets. Herein, a classifying procedure becomes the chief objective and $\alpha\ = 1$.

3.3. DSAE-Based Classification Model

To detect and classify intrusions, the DSAE technique is applied. AE was an unsupervised feature learning NN which comprises hidden states that signify learned features, an input layer that represents input, and an output layer with a similar parameter to the input layer that signifies reconstruction term [23]. The hidden and output layer forms the decoding network that is accountable for the reconstruction of the original input in the learned hidden representation code, while the hidden and input layers in the encoding network are accountable for converting the original input into hidden representation code. For every x^d input vector from dataset $\left\{x^d\right\}_{d=1}^{M}$, the $\hat{x}^d$ reconstructed vector and the h^d representation vector are determined in Equations (7) and (8), correspondingly, whereby f denotes the active function, $W^{(1)}$ and $W^{(2)}$ represent the weight matrixes, and $b^{(1)}$ and $b^{(2)}$ indicate the bias vector. In the SAE algorithm, the sigmoid function is utilized. The $L(x^d, \hat{x}^{\mathrm{d}})$ reconstructed error among x^d and $\hat{x}^d$ is determined in Equation (10). The entire cost function of M instances is determined by Equation (10). The initial term of (10) indicates the reconstructed error of the entire dataset, and the next term is the regularized weight penalty that prevents overfitting by constraining the weight magnitude. The weight decay variable can be represented as λ, the layer count of the network is denoted by n_l, the neuron count in l layer is indicated as sl, and the weight connecting vectors among *i-th* neurons in *the* $l + 1$ layer and j-th neuron in the l layer is referred to $W^{(l)}_{ji}$:

$$h^d = f\left(W^{(1)}x^d + b^{(1)}\right) \tag{7}$$

$$\hat{x}^{\mathrm{d}} = f\left(W^{(2)}h^d + b^{(2)}\right) \tag{8}$$

$$L\left(x^d, \hat{x}^{\mathrm{d}}\right) = \frac{1}{2}\|x^d - \hat{x}^d\|^2 \tag{9}$$

$$J(W,\ b) = [\frac{1}{M}\sum_{d=1}^{M} L\left(x^d, \hat{x}^{\mathrm{d}}\right)] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \tag{10}$$

However, the AE simply copies the input data even though the learned feature representation might perfectly rebuild the original input; the feature is not representative of classification and is redundant to a certain extent. In addition, the DSAE has a greater prospect to further learn representative and abstract features of the input than that of A [23]. Equation (11) illustrates the total cost function of SAE, whereby $J(W,\ b)$ as previously shown in Equation (10), and the next is sparsity penalty term, whereby $\hat{\rho}(g = 1, 2 \cdots e)$ denotes the average activation value of g hidden unit that is determined in Equation (12); ρ refers to an artificially provided smaller value named the sparsity variable, and parameter β indicates the sparsity penalty term utilized for controlling the relative significance among the penalty and reconstructed terms.

$$J_{sparse}(W,\ b) = J(W,\ b) + \beta\sum_{g=1}^{e}\left(\rho\log\frac{\rho}{\hat{\rho}_{\mathrm{g}}} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_g}\right) \tag{11}$$

$$\hat{\rho}_{\mathrm{g}} = \frac{1}{M}\sum_{d=1}^{M} h_g^d \tag{12}$$

The DSAE aim is to future learn sparse and representative features that could remain and extract further data from the input rather than copying the input by minimalizing the cost function $J_{sparse}(W,\ b)$ through the backpropagation (BP) model. The optimum set of variables $W^{(1)}$, $W^{(2)}$, $b^{(1)}$, and $b^{(2)}$ are simultaneously learned through the reduction technique that is named the training model of DSAE.

3.4. Hyperparameter Tuning Using BOA

For optimal modification of the hyperparameters related to the DSAE method, the BOA was applied to it. The BOA is a nature-based inspired optimization technique to resolve optimization problems. It depends on the mating and food-searching behaviors of butterflies (BF) [24]. The basic idea was to utilize the foraging approach of BF in nature which depends on a smaller sensor to determine the mating partner or nectar position. Power exponent (a), stimulus intensity (I), and sensory modality (c) are the three significant terms utilized for modelling the idea of processing and sensing the modality in the foraging process. It should be noted that sensory represents the energy form, whereby modality represents the raw input utilized by the sensor. In this study, the modality covering the fragrance (f) is expressed by a function of physical intensity in the following:

$$f = cI^a \tag{13}$$

In Equation (13), the perceived magnitude of fragrances was indicated as f, the sensory modality is indicated as c, the stimulus intensity is characterized by I, and the power exponent that depends on the modality that is accountable for various levels of absorption is denoted as a. The parameters' a and c values are lies within the interval of zero and one.

The initialization, iteration, and final phases are the three stages of BOA. Initially, the algorithm determines its solution space and the objective function. An initialized population of BF is created where the position is generated randomly in the solution space. Next, several iterations are implemented. Each BF in the solution space moves towards a novel position with the evaluated fitness value. Following, the BF fragrance can be produced at the position using Equation (14). The iteration state is repeated until the

ending condition is satisfied. At last, the algorithm output of the optimal solution obtained the optimal fitness value.

Local and global searching stages are the two major strategies utilized in this study. In a global search, the BF moves towards the solution or fittest BF:

$$x_i^{t+1} = x_i^t + \left(r^2 \times g^* - x_i^t\right) \times f_i \tag{14}$$

In Equation (15), the vector of solution for *i-th* BFs in t iteration is represented as x_i^t, the finest solution found in the existing iteration can be denoted by the term g^*, and the fragrance of *i*-th BFs are denoted as f_i and r, which indicate a random integer between zero and one. Alternatively, the local search technique is modeled through:

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \tag{15}$$

In Equation (15), x_j^t and x_k^t indicate *j*-th and *k*-th BFs from a solution space. The pseudo-code of the BOA has been shown in Algorithm 1. The FF roles are an important part in the BOA system. It computes a positive digit to signify the development of candidate results. The classifier rate of rates has been assumed as FF. The rich solution contains a lower fitness value.

$$\begin{aligned} \textit{fitness}\ (\chi_i) &= \textit{Classifier Error Rate}(\chi_i) \\ &= \frac{\textit{number of misclassified samples}}{\textit{Total number of samples}} * 100 \end{aligned} \tag{16}$$

Algorithm 1: Pseudo-code of BOA

Initialization of (c) sensor modality, (a) power exponent, and (p) switching probability
Determine the fitness function $f(x)$, $x = (x_1, x_2, \ldots, x_{dim})$, dim = dimension count
Produce the initialized population of n BFs
while (end condition is not satisfied) do
for (every bf in the population) do
Evaluate fragrance for bf by utilizing Equation (13)
end for
Discover the optimal bf
for (all the bf in the population) do
Produce a r number randomly from $[0, 1]$
if $r < p$ then
Move to the finest BF by utilizing Equation (14)
else
Randomly move by utilizing Equation (15)
end if
end for
Upgrade the value of a
end while
Output the finest solution

4. Performance Evaluation

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. The performance validation of the IDCPRO-DLM model was performed using the CICIDS2017 dataset (https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 18 December 2022)). It is a network traffic dataset comprising a collection of benign and malicious traffic for network intrusion detection and traffic analysis. The CICIDS2017 dataset is gathered from a realistic network environment that includes traffic records from different protocols, such as TCP, UDP, ICMP, and HTTP. The dataset also includes network traffic records that have been captured from IoT devices, such as smart homes, wearable devices, and industrial control systems. It comprises data samples under diverse attacks such as Brute force attacks, DDoS attacks, normal attacks,

botnet attacks, PortScan attacks, and infiltration attacks. The results are examined under 70% of training data and 30% of testing data.

Table 1 and Figure 2 report the classifier outcomes provided by the IDCPRO-DLM technique on training data. The outcomes implied that the IDCPRO-DLM approach has effectually classified the different kinds of attacks. For instance, the IDCPRO-DLM model has recognized normal attacks with $prec_n$ of 98.65%, $reca_l$ of 98.86%, $spec_y$ of 98.84%, $accu_y$ of 98.07%, and F_{score} of 99.18%. Simultaneously, the IDCPRO-DLM model has recognized botnet attacks with $prec_n$ of 98.54%, $reca_l$ of 98.42%, $spec_y$ of 99.26%, $accu_y$ of 97.94%, and F_{score} of 98.08%. Concurrently, the IDCPRO-DLM model has categorized DDoS attacks with $prec_n$ of 99.17%, $reca_l$ of 98.78%, $spec_y$ of 98.18%, $accu_y$ of 98.41%, and F_{score} of 98.84%. Moreover, the IDCPRO-DLM model has classified PortScan attacks with $prec_n$ of 98.36%, $reca_l$ of 98.54%, $spec_y$ of 98.31%, $accu_y$ of 98.55%, and F_{score} of 98.11%.

Table 1. Result analysis of the IDCPRO-DLM model in the training phase.

Attack Types	Precision (%)	Recall (%)	Specificity (%)	Accuracy (%)	F-Score (%)
Training Phase					
Normal Attack	98.65	98.86	98.84	98.07	99.18
Botnet Attack	98.54	98.42	99.26	97.94	98.08
Brute Force Attack	98.13	97.23	98.52	98.10	98.23
DDoS Attack	99.17	98.78	98.18	98.41	98.84
Infiltration Attack	98.33	98.26	98.59	98.06	97.28
PortScan Attack	98.36	98.54	98.31	98.55	98.11
Average	98.53	98.35	98.62	98.19	98.29

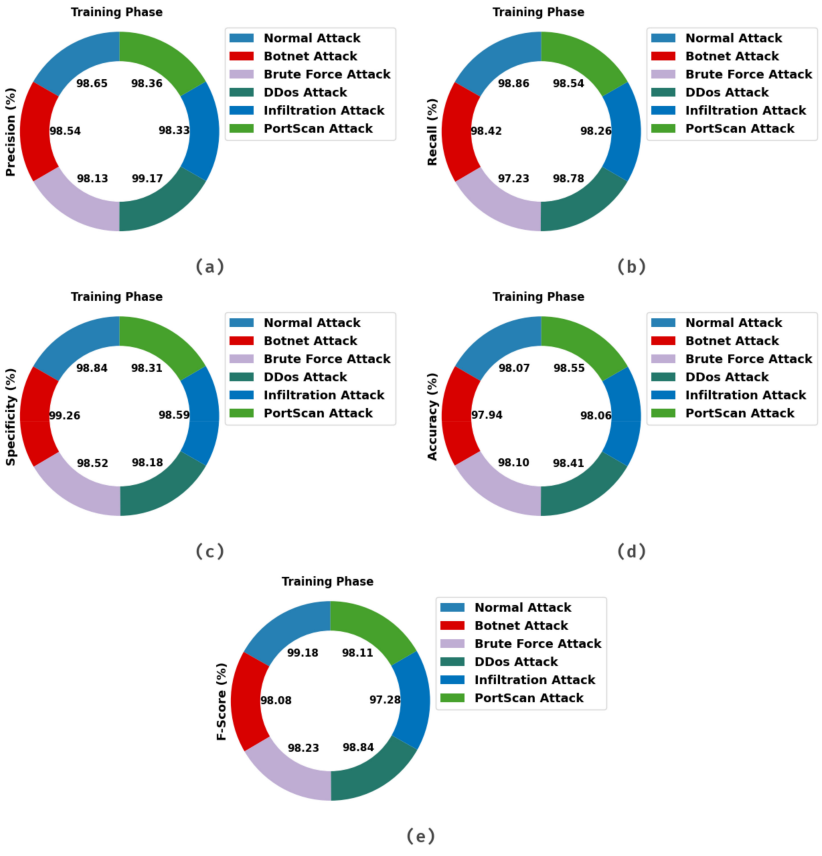


Figure 2. (**a**–**e**) Intrusion detection results of IDCPRO-DLM model on the training phase.

Table 2 and Figure 3 show the classification outcomes given by the IDCPRO-DLM methodology on the testing dataset. The outcome indicates that the IDCPRO-DLM technique has effectively considered the diverse types of attacks. For example, the IDCPRO-DLM approach has identified normal attacks with $prec_n$ of 98.02%, $reca_l$ of 98.51%, $spec_y$ of 98.64%, $accu_y$ of 97.97%, and F_{score} of 98.32%. At the same time, the IDCPRO-DLM method has identified botnet attacks with $prec_n$ of 98.68%, $reca_l$ of 99.08%, $spec_y$ of 99.17%, $accu_y$ of 98.90%, and F_{score} of 98.56%. Simultaneously, the IDCPRO-DLM approach has classified DDoS attacks with $prec_n$ of 98.44%, $reca_l$ of 99.28%, $spec_y$ of 98.54%, $accu_y$ of 97.32%, and F_{score} of 99.26%. Furthermore, the IDCPRO-DLM method has categorized PortScan attacks with $prec_n$ of 99.15%, $reca_l$ of 98.67%, $spec_y$ of 98.86%, $accu_y$ of 98.91%, and F_{score} of 98.41%.

Table 2. Result analysis of the IDCPRO-DLM approach in the testing phase.

Attack Types	Precision (%)	Recall (%)	Specificity (%)	Accuracy (%)	F-Score (%)
Testing Phase					
Normal Attack	98.02	98.51	98.64	97.97	98.32
Botnet Attack	98.68	99.08	99.17	98.90	98.56
Brute Force Attack	98.57	99.00	99.20	99.21	98.75
DDos Attack	98.44	99.28	98.54	97.32	99.26
Infiltration Attack	98.97	98.37	98.02	98.84	98.24
PortScan Attack	99.15	98.67	98.86	98.91	98.41
Average	98.64	98.82	98.74	98.53	98.59

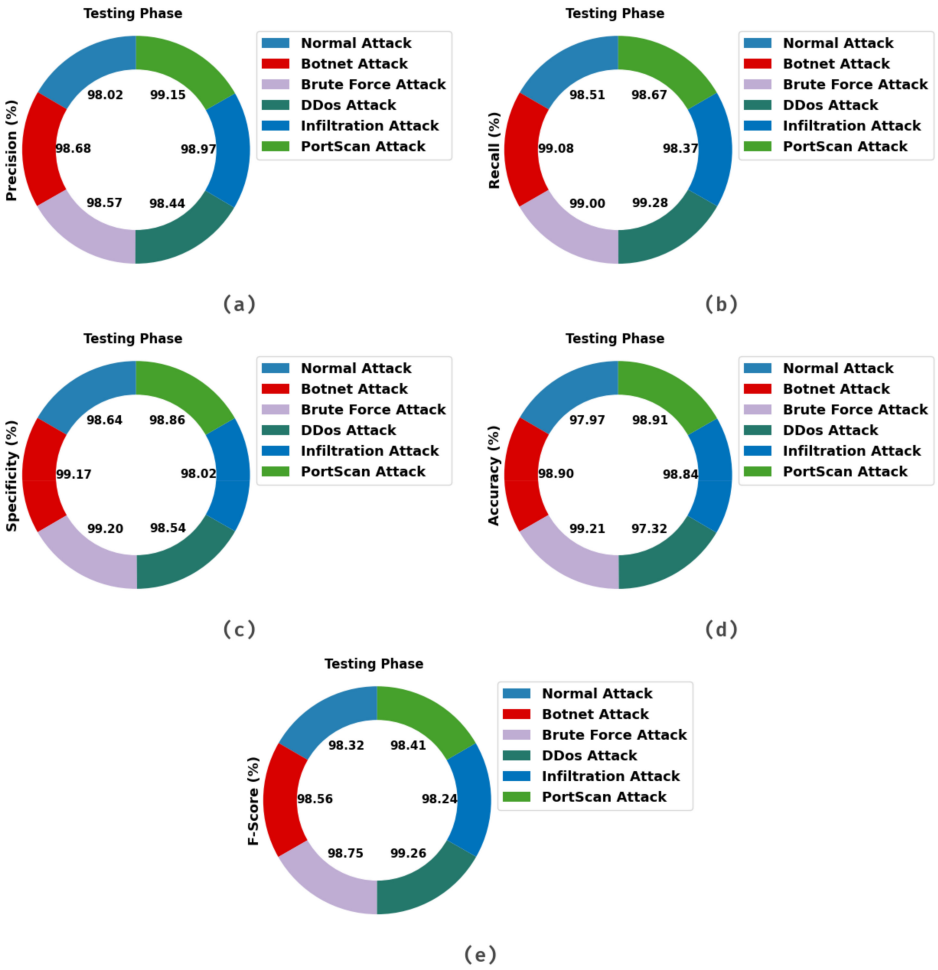


Figure 3. (**a**–**e**) Intrusion detection results of IDCPRO-DLM model in the testing phase.

The TACY and VACY accomplished via the IDCPRO-DLM approach to the testing database are demonstrated in Figure 4. The experimental result showed that the IDCPRO-DLM methodology has obtained higher values of TACY and VACY. Particularly of note, the VACY is higher than the TACY.

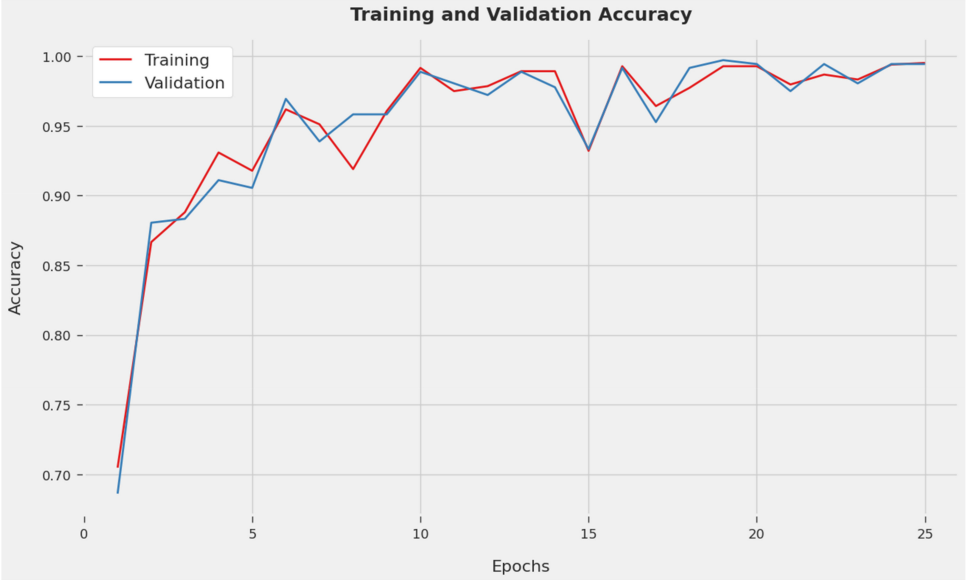


Figure 4. TACY and VACY of IDCPRO-DLM model.

The TLOS and VLOS accomplished via the IDCPRO-DLM methodology on testing data are depicted in Figure 5. The simulation result shows the IDCPRO-DLM method has obtained lesser values of TLOS and VLOS. To be specific, the VLOS is lower than the TLOS.



Figure 5. TLOS and VLOS of IDCPRO-DLM model.

An evident precision–recall analysis of the IDCPRO-DLM system on testing data is demonstrated in Figure 6. The figure demonstrates that the IDCPRO-DLM technique has resulted in improved values of precision–recall values in each class.

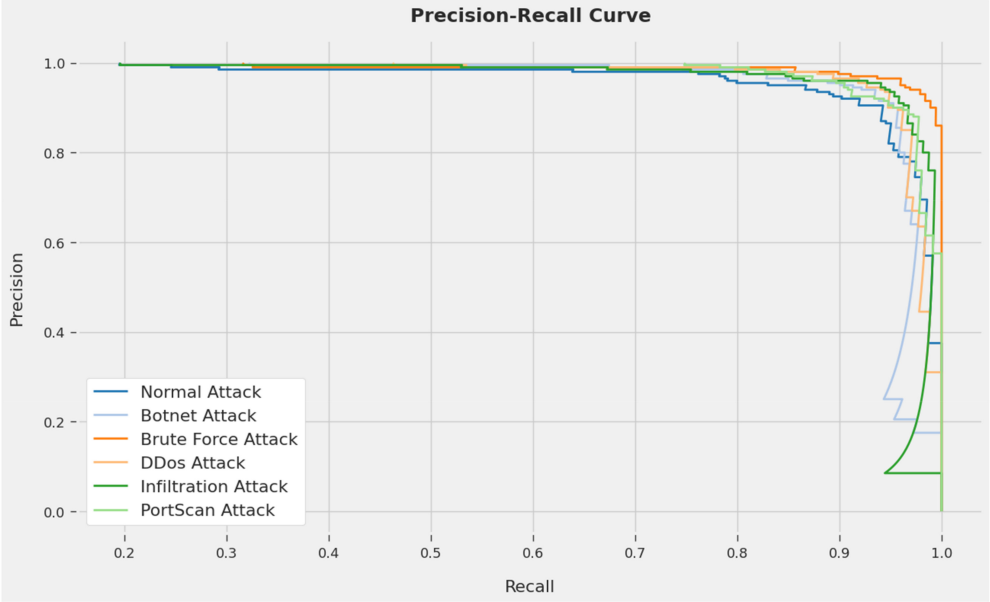


Figure 6. Precision–recall results of IDCPRO-DLM model.

A comprehensive ROC analysis of the IDCPRO-DLM methodology on testing data is demonstrated in Figure 7. The result indicates that the IDCPRO-DLM system has exhibited its capability in classifying dissimilar classes on testing datasets.

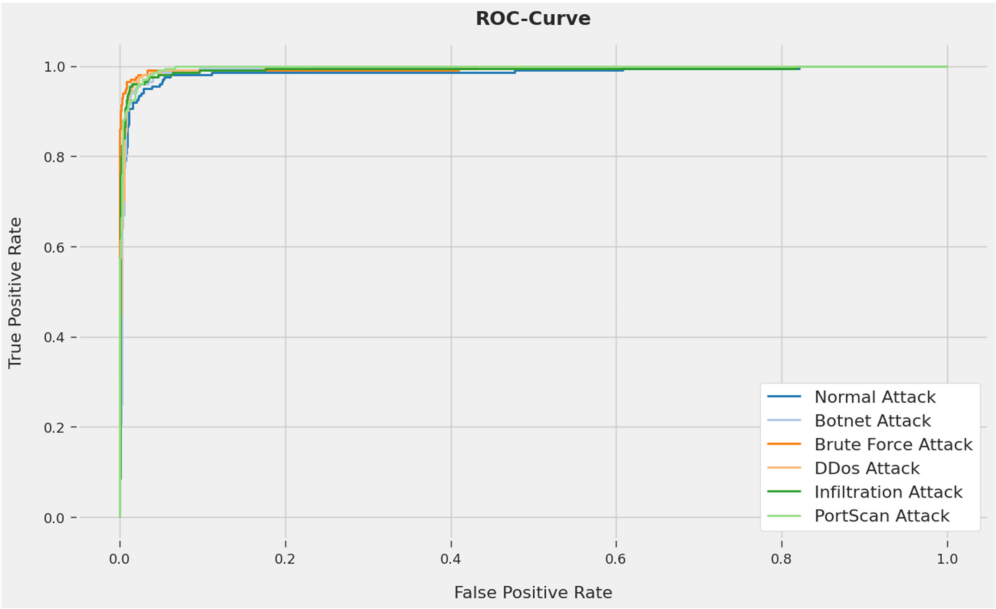


Figure 7. ROC analysis of IDCPRO-DLM model.

To ensure the enhanced performance of the IDCPRO-DLM algorithm, a detailed comparative analysis is made with recent approaches [25] in Table 3. Figure 8 investigates a comparative analysis of the IDCPRO-DLM method with other techniques for $prec_n$ and $reca_l$. The outcomes indicated that the IDCPRO-DLM method has shown increased values of $prec_n$ and $reca_l$. For $prec_n$, the IDCPRO-DLM method has rendered a higher $prec_n$ of 98.64%, whereas the DBN, SVM, RNN, SNN, FFNN, and MDLIDS-SSE models have reported a lower $prec_n$ of 97.80%, 97.05%, 96.96%, 98.14%, 96.26%, and 97.24%, respectively. Additionally, for $reca_l$, the IDCPRO-DLM model has reached a maximum $reca_l$ of 98.82%,

whereas the DBN, SVM, RNN, SNN, FFNN, and MDLIDS-SSE models have attained a minimum $reca_l$ of 96.72%, 96.46%, 97.07%, 97.18%, 96.58%, and 97.91%, respectively.

Table 3. Comparison study of IDCPRO-DLM with existing models [25].

Methods	Precision (%)	Recall (%)	Specificity (%)	Accuracy (%)	F-Score (%)
IDCPRO-DLM	98.64	98.82	98.74	98.53	98.59
Deep Belief Network	97.80	96.72	96.29	98.00	96.96
Support Vector Machine	97.05	96.46	97.29	97.46	97.12
Recurrent Neural Network	96.96	97.07	98.41	96.73	97.31
Spiking Neural Networks	98.14	97.18	96.51	97.02	96.49
Feedforward Neural Network	96.26	96.58	97.85	96.35	97.94
MDLIDS-SSE Model	97.24	97.91	98.26	96.91	96.25

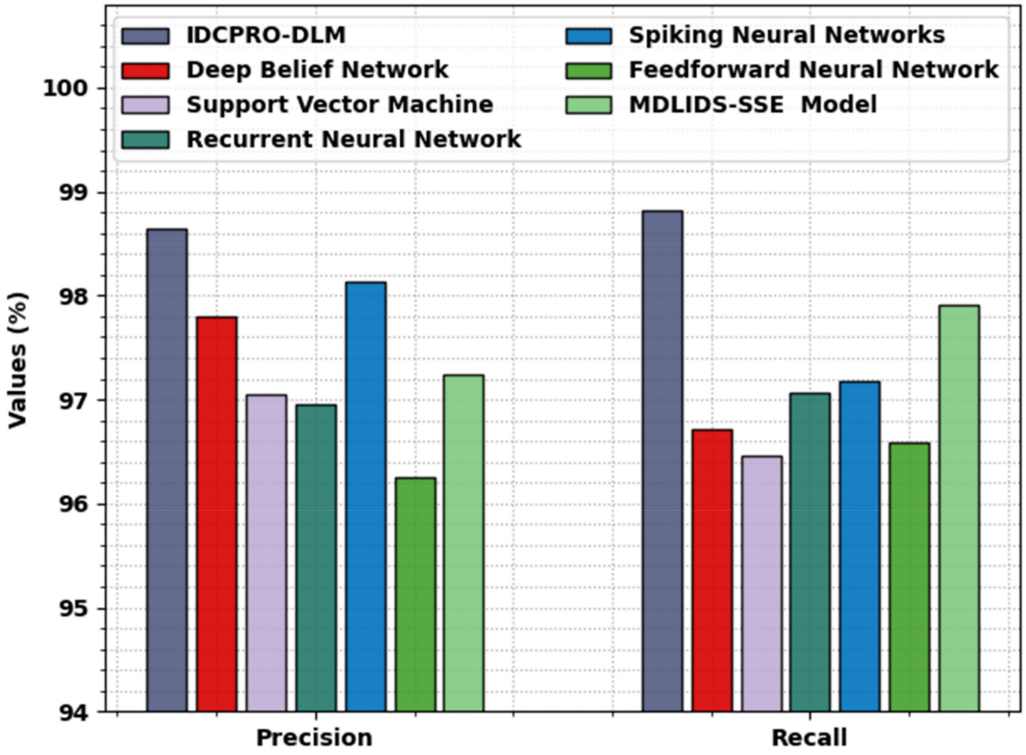


Figure 8. $Prec_n$ and rea_l analysis of IDCPRO-DLM with recent models.

Figure 9 examines a result analysis of the IDCPRO-DLM with existing methods in terms of $spec_y$, F_{score}, and $accu_y$. The result indicates that the IDCPRO-DLM approach has demonstrated higher values of $spec_y$, F_{score}, and $accu_y$. Concerning $spec_y$, the IDCPRO-DLM technique has offered an increased $prec_n$ of 98.74%, while the DBN, SVM, RNN, SNN, FFNN, and MDLIDS-SSE methodologies have exhibited lower $spec_y$ of 96.29%, 97.29%, 98.41%, 96.51%, 97.85%, and 98.26%, correspondingly. As well, concerning $reca_l$, the IDCPRO-DLM approach has reached a maximal $accu_y$ of 98.53%, while the DBN, SVM, RNN, SNN, FFNN, and MDLIDS-SSE approaches have accomplished minimal $accu_y$ of 98%, 97.46%, 96.73%, 97.02%, 96.35%, and 96.91%, correspondingly. Thus, the experimental outcomes showed that the IDCPRO-DLM method has shown maximal performance over other models.

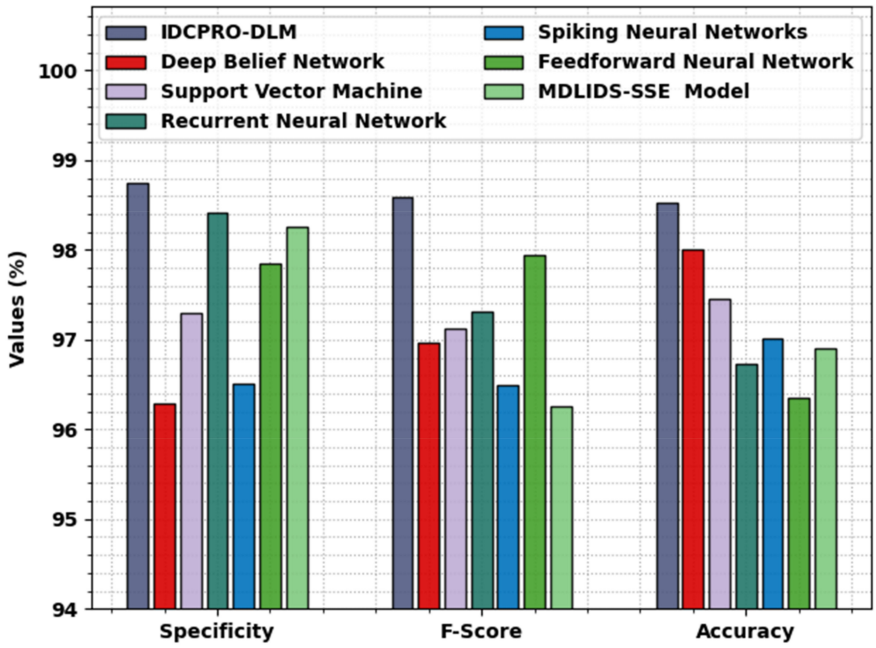


Figure 9. Result analysis of IDCPRO-DLM with recent models.

5. Conclusions

In this article, a novel IDCRPO-DLM approach was devised for the recognition and classification of intrusion in smart environments. The presented IDCRPO-DLM approach incorporates Z-score normalization, CRPOA-based feature subset selection, DSAE-based intrusion classification, and BOA-based hyperparameter tuning. For optimal modification of hyperparameters related to the DSAE method, the BOA was carried out on, which improved the overall classification performance. The simulation analysis of the IDCPRO-DLM approach was tested utilizing the benchmark CICIDS2017 dataset and the comparative results highlighted the betterment of the IDCPRO-DLM algorithm over recent systems with maximum accuracy of 98.53%. In the future, the performance of the IDCPRO-DLM model can be expanded to the design of outlier detection models. In addition, the proposed model can be extended to the detection of Android malware attacks.

Author Contributions: Conceptualization, F.S.A.; Methodology, M.M.A.; Software, A.S.S. and A.S.Z.; Validation, F.S.A., A.S.S. and M.A.H.; Formal analysis, M.I.A.; Investigation, A.S.Z.; Resources, M.M.; Data curation, A.S.S.; Writing—original draft, F.S.A., M.M.A., M.M., M.A.H., S.S.I. and M.I.A.; Writing—review & editing, F.S.A., M.M., A.S.S., S.S.I., A.S.Z. and M.I.A.; Visualization, M.A.H.; Supervision, M.M.A.; Project administration, M.A.H.; Funding acquisition, F.S.A., M.M.A. and M.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under grant number (RGP 2/65/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2023R787), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval of the final version of the manuscript.

References

1. Monteiro, A.C.B.; França, R.P.; Arthur, R.; Iano, Y. A Look at Machine Learning in the Modern Age of Sustainable Future Secured Smart Cities. In *Data-Driven Mining, Learning and Analytics for Secured Smart Cities*; Springer: Cham, Switzerland, 2021; pp. 359–383.
2. Kumar, P.; Gupta, G.P.; Tripathi, R. TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning. *J. Syst. Arch.* **2020**, *115*, 101954. [CrossRef]
3. Tyagi, A.K. Building a smart and sustainable environment using internet of things. In Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Jaipur, India, 26–28 February 2019.
4. Ahad, M.A.; Paiva, S.; Tripathi, G.; Feroz, N. Enabling technologies and sustainable smart cities. *Sustain. Cities Soc.* **2020**, *61*, 102301. [CrossRef]
5. Vijaykumar, J.; Rajkumar, P.; Kandan, S.R. WITHDRAWN: Fog computing based secured mobile cloud for cumulative integrity in smart environment and Internet of Things. *Mater. Today Proc.* **2021**. [CrossRef]
6. Jiang, S.; Zhao, J.; Xu, X. SLGBM: An intrusion detection mechanism for wireless sensor networks in smart environments. *IEEE Access* **2020**, *8*, 169548–169558. [CrossRef]
7. Elrawy, M.F.; Awad, A.I.; Hamed, H.F. Intrusion detection systems for IoT-based smart environments: A survey. *J. Cloud Comput.* **2018**, *7*, 21. [CrossRef]
8. Alsoufi, M.A.; Razak, S.; Siraj, M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 8383. [CrossRef]
9. Keserwani, P.K.; Govil, M.C.; Pilli, E.S.; Govil, P. A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. *J. Reliab. Intell. Environ.* **2021**, *7*, 3–21. [CrossRef]
10. Ramaiah, M.; Chandrasekaran, V.; Ravi, V.; Kumar, N. An intrusion detection system using optimized deep neural network architecture. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4221. [CrossRef]
11. Subbarayalu, V.; Surendiran, B.; Arun Raj Kumar, P. Hybrid network intrusion detection system for smart environments based on internet of things. *Comput. J.* **2019**, *62*, 1822–1839. [CrossRef]
12. Boni, K.R.C.; Xu, L.; Chen, Z.; Baddoo, T.D. A Security Concept Based on Scaler Distribution of a Novel Intrusion Detection Device for Wireless Sensor Networks in a Smart Environment. *Sensors* **2020**, *20*, 4717. [CrossRef]
13. Manimurugan, S.; Al-Mutairi, S.; Aborokbah, M.M.; Chilamkurti, N.; Ganesan, S.; Patan, R. Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network. *IEEE Access* **2020**, *8*, 77396–77404. [CrossRef]
14. Efstathopoulos, G.; Grammatikis, P.R.; Sarigiannidis, P.; Argyriou, V.; Sarigiannidis, A.; Stamatakis, K.; Angelopoulos, M.K.; Athanasopoulos, S.K. Operational data based intrusion detection system for smart grid. In Proceedings of the 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11–13 September 2019; pp. 1–6.
15. Al Razib, M.; Javeed, D.; Khan, M.T.; Alkanhel, R.; Muthanna, M.S.A. Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework. *IEEE Access* **2022**, *10*, 53015–53026. [CrossRef]
16. Saheed, Y.K.; Arowolo, M.O. Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms. *IEEE Access* **2021**, *9*, 161546–161554. [CrossRef]
17. Moradi, M.; Moradkhani, M.; Tavakoli, M.B. Enhancing Security on Social Networks with IoT-based Blockchain Hierarchical Structures with Markov Chain. *J. Adv. Comput. Res.* **2022**, *13*, 1–26.
18. Fatani, A.; Dahou, A.; Al-Qaness, M.A.A.; Lu, S.; Elaziz, M.A. Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System. *Sensors* **2021**, *22*, 140. [CrossRef]
19. Fatani, A.; Elaziz, M.A.; Dahou, A.; Al-Qaness, M.A.A.; Lu, S. IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization. *IEEE Access* **2021**, *9*, 123448–123464. [CrossRef]
20. D'hooge, L.; Verkerken, M.; Wauters, T.; De Turck, F.; Volckaert, B. Investigating Generalized Performance of Data-Constrained Supervised Machine Learning Models on Novel, Related Samples in Intrusion Detection. *Sensors* **2023**, *23*, 1846. [CrossRef]
21. Khoei, T.T.; Kaabouch, N. A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems. *Information* **2023**, *14*, 103. [CrossRef]
22. Moosavi, S.H.S.; Bardsiri, V.K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **2019**, *86*, 165–181. [CrossRef]
23. Sun, W.; Shao, S.; Zhao, R.; Yan, R.; Zhang, X.; Chen, X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement* **2016**, *89*, 171–178. [CrossRef]
24. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2018**, *23*, 715–734. [CrossRef]
25. Malibari, A.A.; Alotaibi, S.S.; Alshahrani, R.; Dhahbi, S.; Alabdan, R.; Al-wesabi, F.N.; Hilal, A.M. A novel metaheuristics with deep learning enabled intrusion detection system for secured smart environment. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102312. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.