electronics

MDPI

Article

Intelligent Intrusion Detection Using Arithmetic Optimization Enabled Density Based Clustering with Deep Learning

Fadwa Alrowais [1], Radwa Marzouk [2], Mohamed K. Nour [3], Heba Mohsen [4], Anwer Mustafa Hilal [5,*], Ishfaq Yaseen [5], Mohamed Ibrahim Alsaid [5] and Gouse Pasha Mohammed [5]

1 Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
2 Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
3 Department of Computer Science, College of Computing and Information System, Umm Al-Qura University, Mecca 24381, Saudi Arabia
4 Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt
5 Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
* Correspondence: a.hilal@psau.edu.sa

Abstract: Rapid advancements in the internet and communication domains have led to a massive rise in the network size and the equivalent data. Consequently, several new attacks have been created and pose several challenging issues for network security. In addition, the intrusions can launch several attacks and can be handled by the use of intrusion detection system (IDS). Though several IDS models are available in the literature, there is still a need to improve the detection rate and decrease the false alarm rate. The recent developments of machine learning (ML) and deep learning (DL)-based IDS systems are being deployed as possible solutions for effective intrusion detection. In this work, we propose an arithmetic optimization-enabled density-based clustering with deep learning (AOEDBC-DL) model for intelligent intrusion detection. The presented AOEDBC-DL technique follows a data clustering process to handle the massive quantity of network data traffic. To accomplish this, the AOEDBC-DL technique applied a density-based clustering technique and the initial set of clusters are initialized using the arithmetic optimization algorithm (AOA). In order to recognize and classify intrusions, a bidirectional long short term memory (BiLSTM) mechanism was exploited in this study. Eventually, the AOA was applied as a hyperparameter tuning procedure of the BiLSTM model. The experimental result analysis of the AOEDBC-DL algorithm was tested using benchmark IDS datasets. Extensive comparison studies highlighted the enhancements of the AOEDBC-DL technique over other existing approaches.

Keywords: security; intrusion detection systems; data clustering; deep learning; metaheuristics

Citation: Alrowais, F.; Marzouk, R.; Nour, M.K.; Mohsen, H.; Hilal, A.M.; Yaseen, I.; Alsaid, M.I.; Mohammed, G.P. Intelligent Intrusion Detection Using Arithmetic Optimization Enabled Density Based Clustering with Deep Learning. *Electronics* **2022**, *11*, 3541. https://doi.org/10.3390/electronics11213541

Academic Editor: Vijayakumar Varadarajan

Received: 5 October 2022
Accepted: 26 October 2022
Published: 30 October 2022

1. Introduction

Nowadays, data security and privacy have become a primary concern, and intrusion detection systems (IDSs) play an essential part in cybersecurity [1]. The Industry 4.0 ecosystem is capable of collecting information, interconnecting between one another, and processing and deciding without human intervention [2]. At present, the amount of data traveling via network is overwhelming in terms of the velocity of the Internet links, the volume of the information variety, and veracity of the information that is transferred [3,4]. The traditional IDS system uses the signature-based technique which assists in identifying known attacks and securing the network. However, they still suffer from reduced recognition performance and an increased false alarm rate [5].

Typically, IDS categorizes abnormal traffic as misuse and anomaly-based methods, with all the classes having benefits and drawbacks [6]. The anomaly-based method collects

information that characterizes normal behavior and constructs a familiarity model, and actions deviated from the model are labeled as anomalous or suspicious. At the same time, the misuse-based method compares the data against a predetermined set of patterns or rules to identify network attacks [7]. However, this approach is not adaptable and is constrained in its capability to identify formerly unnoticed attack types [8]. Although a promising improvement accomplished by earlier studies, IDS is still a challenge. This can be exacerbated by an abundance of available features, the higher amount of traffic datasets, and a continuously evolving environment. Fortunately, machine learning (ML) techniques could assist in solving major tasks involving classification, regression, and prediction [9]. The ML technique has been efficiently applied in various applications in intelligent IDSs involving malware detection, network traffic analysis, access logs analysis, and spam. IDS is a proactive network security defense technique that could make up for the limitations of conventional static security policy and thus become a reasonable complement to conventional static defense approaches such as firewalls [10]. System administrator security management abilities were expanded via monitoring, auditing, and response, and reduced the workloads of system administrators.

This study emphasizes the proposal of an arithmetic optimization-enabled density-based clustering with deep learning (AOEDBC-DL) model for intelligent intrusion detection. The presented AOEDBC-DL technique made use of a density-based clustering technique and the initial set of clusters were selected using the arithmetic optimization algorithm (AOA). For intrusion detection, a bidirectional long short-term memory (BiLSTM) model was employed. Eventually, the quantum bat algorithm (QBA) was applied as a hyperparameter tuning procedure of the BiLSTM model. The experimental result analysis of the AOEDBC-DL algorithm was tested using benchmark IDS datasets. In short, the paper contribution is summarized as follows.

- An intelligent AOEDBC-DL model encompassing density based clustering, AOA based initial cluster set selection, BiLSTM intrusion detection, and QBA based hyperparameter tuning is presented for intrusion detection. To the best of our knowledge, the presented AOEDBC-DL model does not exist in the literature;
- AOA was derived with a density-based clustering technique to group the data points into a cluster and the AOA was used for optimal selection of initial cluster points;
- A new QBA was designed with a BiLSTM model for intrusion detection and the choice of QBA helped to appropriately select the hyperparameters of the BiLSTM model;
- The performance of the AOEDBC-DL model was validated on the WSN-DS (Wireless Sensor Networks-Dataset) dataset, which contains 15,000 samples with five class labels, namely normal, blackhole, gray hole, flooding, and scheduling attacks.

The rest of the paper is organized as follows. Section 2 offers a brief set of existing works and Section 3 introduces the proposed AOEDBC-DL model. Section 4 provides experimental validation and Section 5 draws the concluding remarks.

2. Literature Review

Albulayhi et al. [11] implemented and proposed a novel extraction approach and feature selection (FS) for anomaly-based IDS. The method starts by utilizing two entropy-based concepts (gain ratio (GR) and information gain (IG)) to extract and select appropriate characteristics in different ratios. Next, union and intersection mathematical concepts are used for extracting better features. Reference [12] proposed a robust wrapper FS methodology for decreasing the processing time and enhancing the performance of the IDS. The presented technique exploits a DE model for selecting the relevant features whereas the ELM (Extreme Learning Machine) classifiers are used after FS to calculate the selected feature. Li et al. [13] developed a powerful DL methodology such as AE-IDS (Auto-Encoder Intrusion Detection) based on an RF (radio frequency) mechanism. After training, it forecasts the outcomes with AE that effectually enhance the predictive performance and decrease the detection time.

Gopalakrishnan and Purusothaman [14] proposed a Higher Ranking-based Optimized ELM (HR-OELM) based on three dissimilar classifiers to develop a smart IDS. The most important highlight of the optimum FS is to decrease the correlation amongst the features by providing unique data. This feature was subjected to the presented algorithm where the Adaboost, DNN (deep neural network), and RF classifications were used. The recognition accuracy can be concluded according to the higher ranking of output from three classifications. In [15], the authors developed a hybrid ML method termed XGB-RF for IDS attacks. The presented technique was employed for the N-BaIoT (Network-based Detection of IoT) data, encompassing hazardous botnet attacks. RF was utilized for the XGBoost (eXtreme Gradient Boosting) and FS classifiers have been utilized for detecting various kinds of attacks on the IoT environment.

Upadhyay et al. [16] introduced a comprehensive structure for the IDS for smart grids that integrates feature engineering-based pre-processing with ML classifiers. Although the ML technique fine-tunes the hyper-parameter to increase the recognition rate, the presented method focused on choosing the promising features of the data through Gradient Boosting FS (GBFS) before employing the classification model, an integration that enhances the execution speed and the recognition rate. GBFS makes use of the Weighted Feature Importance (WFI) extraction model for decreasing the classifier complexity. In [17], the authors proposed a hybrid and layered IDS that employs an integration of dissimilar ML and FS algorithms for providing improved performance in distinct kinds of attacks. Two new techniques were developed for the FS process. The layered structure is generated by defining a proper ML algorithm based on the types of attack.

Ullah et al. [18] proposed a hybrid deep learning (DL) technique to identify cyberattacks in the IoV. The presented technique mainly depends upon long short-term memory (LSTM) and gated recurrent unit (GRU). The experimental validation of the presented model can be examined on two benchmark datasets. Otoum et al. [19] combined the spider monkey optimization (SMO) and stacked-deep polynomial network (SDPN) for attaining an optimum detection process. The SMO algorithm chooses the optimum features in the dataset and SDPN performs an anomaly classification process. In [20], the major cyberattacks can be determined by the use of a DL model. Several security features such as DoS, malevolent operation, data type probing, spying, scanning, intrusion detection, brute force, web attacks, and wrong setup were examined by the use of a sparse evolutionary training (SET)-based classification model. Nasir et al. [21] introduced a new DF-IDS (Dataframe-Intrusion detection systems) for intrusion detection in the IoT traffic. It includes two major stages. Initially, the selection of features from the feature matrix is performed by the use of Spider Monkey (SM), principal component analysis (PCA), information gain (IG), and correlation attribute evaluation (CAE). Next, the features with integrated labels can be employed for training the deep neural network to identify intrusions.

3. The Proposed Model

In this work, we have developed a new AOEDBC-DL model for intelligent intrusion detection. Initially, the network data were preprocessed by the AOEDBC-DL technique, where the clustering process took place to organize the network data into several clusters by the use of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) model. Then, the clustered data were passed into the BiLSTM model where the intrusions were identified properly. To improve the intrusion detection performance, the hyperparameters related to the BiLSTM model can be optimally chosen by the QBA. Figure 1 illustrates the overall process of the AOEDBC-DL system.

3.1. Data Clustering Using DBSCAN Model

The AOEDBC-DL technique applied density based clustering algorithm to group the network data. In the presented method, the DBSCAN algorithm was employed for dividing the data into clusters of similar characteristics. The presented model is a non-parametric spatial clustering model [22]. The key conception of the DBSCAN is that for all the clusters,

the point count in eps distance is better than the threshold density. When the recovered neighborhood holds minimal ***minPts*** points, a novel cluster, ***C*** is appended. The procedure can be demonstrated in Algorithm 1.

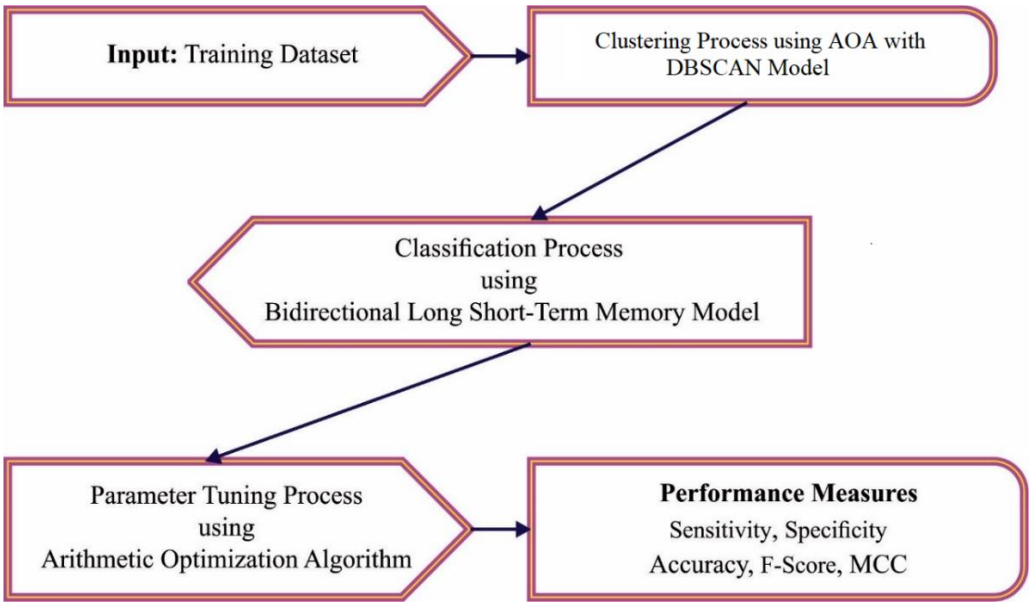

Figure 1. Overall process of AOEDBC-DL system.

Algorithm 1: DBSCAN Algorithm

Input: distance ϵ, Dataset D, minimal cluster density ***minPts***
Begin
$C < -0$
For all the ***P*** points in D, data do
  If ***P*** is visited then
    Carry out subsequent ***P***
  Else
    Set ***P*** as visited
    nbrPts <- points in neighborhood of ***P***
  End if
  If $|nbrPts| < minPts$ then
    Set ***P*** as Noise
  Else
$C < -Newcluster$
    Implement Expand_Cluster_Function (***P***, ***nbrPts***, ***C***, ***minPts***)
  End if
End for

In this study, the initial set of clusters was initialized using the AOA. AOA is a novel variety of a swarm intelligence (SI) technique presented by Mirjalili in 2020 [23]. This technique is an easy infrastructure, has some parameters, and is simple for execution. Its searching method is mostly controlled by fundamental mathematical operators such as subtraction (S " $-$ "), division (D " $\div$ "), addition (A " $+$ "), and multiplication (M" $\times$ "). Primarily, AOA has been obtained by generating several primary arbitrary candidate solutions $\in (x_1, x_2, x_3, \ldots, x_n)$. Before the exploration step, Math Optimizer Accelerated (MOA) can be estimated. Afterward, the AOA system can be established; it is the first to enter the exploration step. MOA has been reached by the subsequent Equation (1):

$$MOA(C_{Iter}) = \text{Min} + C_{Iter} \times \left(\frac{\text{Max} - \text{Min}}{M_{Iter}} \right), \quad (1)$$

where C_{Iter} stands for the current iteration $C_{Iter} \in (1,\ M_{Iter})$. Max and Min denote the maximal and minimal values of acceleration functions. $MOA(C_{Iter})$ implies the function value in tth th iteration that is attained by Equation (2):

$$x_{ij}(C_{Iter}+1) = \begin{cases} best(\chi_j) \div (MOP+e) \times \left(\left(ub_j - lb_j\right) \times \mu + lb_j\right), r_2 < 0.5 \\ best(xj) \times MOP \times \left(\left(ub_j - lb_j\right) \times \mu + lb_j\right),\ r_2 \geq 0.5 \end{cases}, \quad (2)$$

in which $x_i(C_{Iter}+1)$ defines the ith solution from the next iteration, $x_{ij}(C_{Iter})$ signifies the jth position of ith solution from the current iteration, $best(xj)$ represents the jth position from the optimum iteration e refers to the smaller integer number, ub_j and lb_j indicates the upper and lower limits of jth position correspondingly, $\mu = 0.5$. r_1, r_2, *and* r_3 represents a random number produced in a range of [0, 1]. Based on the division (D) or multiplication (M) operators, they gain a distributing value or decision. These make it easy for the process exploration mechanism, by employing multiplication (M) or division (D) as follows:

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{\frac{1}{\alpha}}}{M_{Iter}^{\frac{1}{\alpha}}}, \quad (3)$$

in which C_{Iter} implies the existing iteration, and (M_{Iter}) denotes the maximal count of iterations. α implies the sensitive parameter that determines the increased accuracy in the iterative procedure; in this work, $\alpha = 5$.

$$x_{ij}(C_{Iter}+1) = \begin{cases} best(x_j) + MOP \times \left(\left(ub_j - lb_j\right) \times \mu + lb_j\right), & r_3 \geq 0.5 \\ best(\chi_{j_b} - MOP \times \left(\left(ub_j - lb_j\right) \times \mu + lb_j\right), & r_3 < 0.5 \end{cases}. \quad (4)$$

3.2. Intrusion Detection Using Optimal BiLSTM Model

To recognize intrusions, the BiLSTM model was utilized in this study. LSTM (Long short-term memory) can be established by a certain memory cell to store temporary data [24]. This infrastructure allows LSTM to recall longer-range features superior to typical RNN (recurrent neural network). Utilizing multi-layer methods, elements of cells at time step i at l layers from the forward direction were implemented as:

$$f_i^l = \sigma(W_{(f)}^l \overrightarrow{h}_{\mathrm{i}}^{l-1} + V_{(f)}^l \overrightarrow{h}_{\mathrm{i}-1}^{l} + b_{(f)}^l), \quad (5)$$

$$i_i^l = \sigma(W_{(i)}^l \overrightarrow{h}_{\mathrm{i}}^{l-1} + V_{(i)}^l \overrightarrow{h}_{\mathrm{i}-1}^{l} + b_{(i)}^l), \quad (6)$$

$$0_i^l = \sigma(W_{(0)}^l \overrightarrow{h}_{\mathrm{i}}^{l-1} + V_{(0)}^l \overrightarrow{h}_{\mathrm{i}-1}^{l} + b_{(0)}^l), \quad (7)$$

$$g_i^l = \tanh\left(W_{(g)}^l \overrightarrow{h}_{\mathrm{i}}^{l-1} + V_{(g)}^l \overrightarrow{h}_{\mathrm{i}-1}^{l} + b_{(g)}^l\right), \quad (8)$$

$$C_i^l = f_i^l \odot C_{i-1}^l + i_i^l \odot g_i^l, \quad (9)$$

$$\overrightarrow{h}_i^l = 0_i^l \odot \tanh\left(C_i^l\right), \quad (10)$$

in which f_i^l, $\overrightarrow{h}_i^l$, i_i^l, 0_i^l, C_i^l, and g_i^l stand for the forget gate, hidden layer, input gate, output gate, cell state, and candidate gate, correspondingly. In Equations (6)–(9), W^l demonstrates the weighted matrices betwixt cell layers $(l-1)-l$, V^l represents the weighted matrices betwixt consecutive cells of layer l; also b^l defines the bias vector at every layer. The bias value and weighted matrix from the cells were distributed with length of series, thus reducing the entire count of hidden neurons and weighted from the networks. The sigmoid function σ and hyperbolic tangent function were utilized as activation functions and $\odot$ implies the elementwise multiplication.

A BiLSTM is a procedure of the data from backwards as well as forwards directions with two varying LSTM layers. The forward hidden state, $\overrightarrow{h}_i^l$, evaluated employing the above formula, and the backward states, $\overleftarrow{h}_i^l$, are concatenated, and then provided as to the subsequent layers:

$$\overleftrightarrow{h}_i^l = \begin{bmatrix} \overrightarrow{h}_i^l, \\ \overleftarrow{h}_i^l \end{bmatrix}, \tag{11}$$

in which $l = 0$ represents the input layer. BDLSTM (Bidirectional Long Short-Term) is improved at obtaining the correlation betwixt the elements in a total series with data from both directions, alternatively recalling the feature from one direction. Additionally, with the parameter-shared approach, the BDLSTM techniques need minimal memory to resolve the issues compared to the typical CNN and FNN (Fuzzy Neural Network) approaches. Figure 2 depicts the framework of BiLSTM.

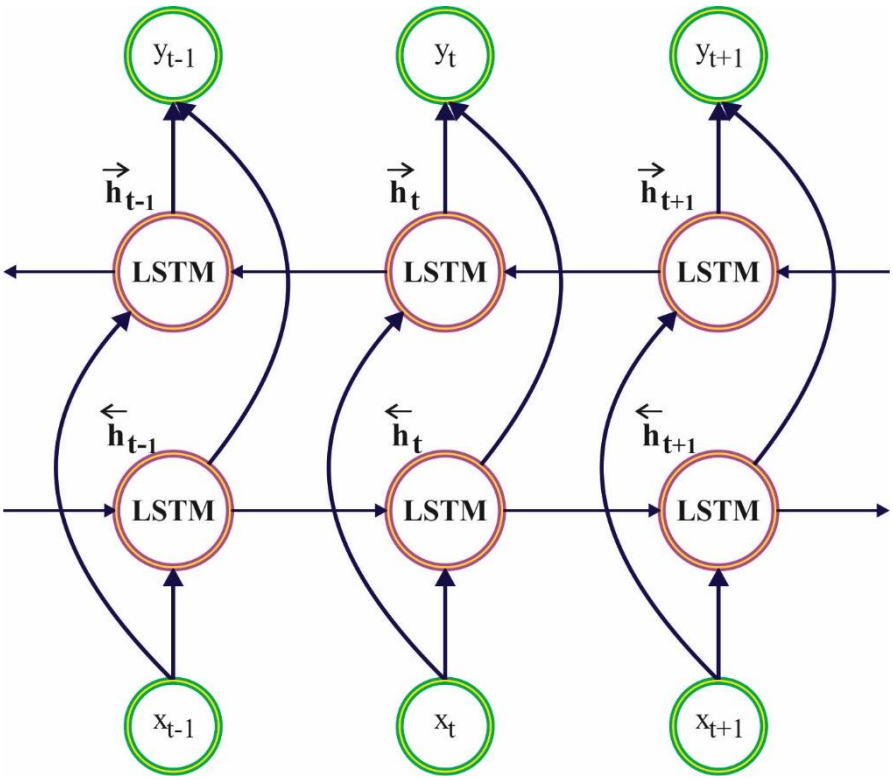

Figure 2. Architecture of BiLSTM.

Lastly, the QBA was applied as a hyperparameter tuning procedure of the BiLSTM model. Because of the higher exploration rate, it was utilized to estimate the search space. Yang [25] developed the BA and created the BA based on three rules. Firstly, they stated that the use of echolocation ability in each bat is similar, and echolocation ability realizes the distances between prey (food) and various background barriers. Next, bat in the x_i location has velocity v_i with fixed frequency $f_{\min}$ and differing wavelength λ_0 uses loudness A_0 to find food. The upper and lower boundaries are utilized for initializing the bat location as follows:

$$X_{ij} = X_0 - (X_m - X_0)rand. \tag{12}$$

In Equation (12), X_{ij} represents the location of jth parameter of ith bat, X_0 and X_m represent the upper and lower boundaries, correspondingly, and *rand* indicates a random value within [0, 1] as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\alpha; \tag{13}$$

$$v_i^t = v_i^{t-1} + (x_i^t - g^t) f_i; \tag{14}$$

$$x_i^t = x_i^{t-1} + v_i^t; \tag{15}$$

where α characterizes a random value in [0, 1], f_i signifies the pulse frequency, $f_{\min}$ shows the minimal frequency, and $f_{\max}$ indicates the maximal frequency. Moreover, g^t shows the global optimum location. x_i^t and x_i^{t-1} depicts the ith bats location at t and $(t - 1)$ iterations, correspondingly. v_i^t and v_i^{t-1} denote the ith velocity for t and the $(t - 1)$ iterations, correspondingly. The Doppler Effect is taken into account as follows:

$$f_{id} = \frac{(340 + v_i^{t-1})}{(340 + v_g^{t-1})} \times f_{id} \times \left[1 + C_i \times \frac{(g_d^t - x_{id}^t)}{|g_d^t - x_{id}^t| + \varepsilon}\right] \tag{16}$$

$$v_{id}^t = (w \times v_{id}^{t-1}) + (g_d^t - x_{id}^t) f_{id}, \tag{17}$$

$$x_{id}^t = x_{id}^{t-1} + v_{id}^t, \tag{18}$$

where f_{id} signifies the ith bat frequency at d dimension, v_g^{t-1} and v_g^t signify the velocity for the global optimal location at $(t-1)$th and tth iterations, and C_i shows the number ranges within [0, 1]. ε is introduced; hence σ^2, the standard deviation, remains positive. Moreover, w represents the weight, g_d^t represents the location in the d dimension for the global optima of t iteration. x_{id}^t symbolizes the location in the d dimension for ith bat at t iteration, x_{id}^{t-1} signifies the location in the d dimension for ith bat at $t - 1$ iteration, v_{id}^t represents the velocity in the d dimension for ith bat at t iteration, v_{id}^{t-1} indicates the velocity in d dimension for ith bat at $t - l$ iteration, [26]:

$$x_{id}^{t+1} = g_d^t \cdot \left[1 + j(0,\ \sigma^2)\right] \sigma^2 = \left|A_i^t - A^t\right| + \varepsilon. \tag{19}$$

In Equation (19), x_{id}^{t+1} shows the ith bat's location in the d dimension at $t + 1$ iteration, $j(0, \sigma^2)$ represents a Gaussian distribution with zero mean and a standard deviation of σ^2, and g_d^t indicates the global optimum location in d dimension at $t + 1$ iteration. A_i^t refers to the ith bat's loudness. Equation (19) illustrates that the global optimum g_d^t is an attractant as follows:

$$x_{id}^t = g_d^t + \beta \left|mbest_d - x_{id}^t\right| ln\left(\frac{1}{u}\right),\ u(0,\ 1) < 0.5; \tag{20}$$

$$x_{id}^t = g_d^t - \beta \left|mbest_d - x_{id}^t\right| ln\left(\frac{1}{u}\right),\ u(0,\ 1) \geq 0.5. \tag{21}$$

Now, u represents the random integer. β indicates the contraction coefficient, x_{id}^t denotes the ith bat's location in the d dimension for t iteration, and $mbest_d$ shows the average personal best in d dimension.

After formalization of a novel solution, we carefully chose various solutions and applied a random walk as:

$$x_n = x_o + \varepsilon A^t. \tag{22}$$

In Equation (22), A^t symbolizes the average loudness of bats, ε represents a random integer, x_o signifies the existing position, and x_n indicates the novel location afterward the local search. In all iterations, the subsequent equation is updating the loudness A_i and pulse rate r_i:

$$A_i^{t+1} = \triangle A_i^t, \tag{23}$$

$$r_i^{t+1} = r_i^0 \left[1 - \exp\left(-\frac{\gamma}{t}\right)\right], \tag{24}$$

where A_i^{t+1} denotes the ith bat's loudness in $(t + 1)$th iteration and A_i^t signifies the ith bat's loudness in tth iteration. r_i^0 represents the ith bat's early pulse rate, γ and $\triangle$ denotes constant value, and r_i^{t+1} symbolizes the ith bat's pulse rate at $(t + 1)$th iteration.

4. Performance Validation

In this section, the intrusion detection results of the AOEDBC-DL model were studied on the WSN-DS dataset [27]. The dataset includes 15,000 samples with five class labels as represented in Table 1. Each class holds a set of 3000 samples.

Table 1. Dataset details.

Label	Class	No. of Samples for Experiment
C-1	Normal	3000
C-2	Blackhole	3000
C-3	Grayhole	3000
C-4	Flooding	3000
C-5	Scheduling Attacks	3000
Total Number of Samples		**15,000**

The confusion matrices attained by the AOEDBC-DL model on the test WSN-DS dataset under different experiments are depicted in Figure 3. The AOEDBC-DL model has categorized all the classes accurately and automatically. For instance, with Experiment-1, the AOEDBC-DL model has recognized 19.22% of samples under C-1, 19.39% of samples under C-2, 19.55% of samples under C-3, 19.23% of samples under C-4, and 19.49% of samples under C-5. Next to that, with Experiment-3, the AOEDBC-DL method has identified 19.75% of samples under C-1, 19.79% of samples under C-2, 19.78% of samples under C-3, 19.85% of samples under C-4, and 19.72% of samples under C-5. At last, with Experiment-5, the AOEDBC-DL approach has categorized 19.63% of samples under C-1, 19.68% of samples under C-2, 19.64% of samples under C-3, 19.78% of samples under C-4, and 19.61% of samples under C-5.

Figure 3. *Cont.*

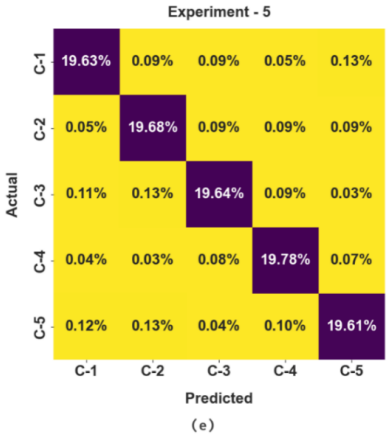

Figure 3. Confusion matrices of AOEDBC-DL system (**a**) Experiment-1, (**b**) Experiment-2, (**c**) Experiment-3, (**d**) Experiment-4, and (**e**) Experiment-5.

The intrusion recognition results of the AOEDBC-DL model are examined in Table 2 and Figure 4. The outcomes established that the AOEDBC-DL model recognized the intrusions proficiently. For example, in Experiment-1, the AOEDBC-DL model offered an average $accu_y$ of 98.75%, $sens_y$ of 96.88%, $spec_y$ of 99.22%, F_{score} of 96.88%, and MCC of 96.10%. Concurrently, in Experiment-3, the AOEDBC-DL algorithm has gained an average $accu_y$ of 99.55%, $sens_y$ of 98.88%, $spec_y$ of 99.72%, F_{score} of 98.88%, and MCC of 98.60%. Simultaneously, in Experiment-4, the AOEDBC-DL system has gained an average $accu_y$ of 99.70%, $sens_y$ of 99.25%, $spec_y$ of 99.81%, F_{score} of 99.25%, and MCC of 99.07%. Finally, in Experiment-5, the AOEDBC-DL algorithm has accomplished an average $accu_y$ of 99.34%, $sens_y$ of 98.35%, $spec_y$ of 99.59%, F_{score} of 98.35%, and MCC of 97.93%.

Table 2. Result analysis of AOEDBC-DL system with distinct measures and class labels.

Labels	Accuracy	Sensitivity	Specificity	F_{score}	MCC
			Experiment-1		
C-1	98.59	96.10	99.21	96.45	95.57
C-2	98.73	96.93	99.18	96.82	96.02
C-3	98.91	97.73	99.21	97.30	96.62
C-4	98.89	96.17	99.58	97.20	96.52
C-5	98.64	97.47	98.93	96.63	95.78
Average	**98.75**	**96.88**	**99.22**	**96.88**	**96.10**
			Experiment-2		
C-1	99.37	98.63	99.55	98.42	98.02
C-2	99.41	98.73	99.58	98.52	98.15
C-3	99.46	98.67	99.66	98.65	98.31
C-4	99.51	98.70	99.71	98.77	98.46
C-5	99.33	97.93	99.68	98.31	97.89
Average	**99.41**	**98.53**	**99.63**	**98.53**	**98.17**
			Experiment-3		
C-1	99.51	98.73	99.71	98.78	98.48
C-2	99.53	98.93	99.68	98.82	98.52
C-3	99.61	98.90	99.79	99.03	98.79
C-4	99.61	99.23	99.70	99.02	98.77
C-5	99.50	98.60	99.72	98.75	98.44
Average	**99.55**	**98.88**	**99.72**	**98.88**	**98.60**

Table 2. *Cont.*

Labels	Accuracy	Sensitivity	Specificity	F_{score}	MCC
			Experiment-4		
C-1	99.71	98.80	99.94	99.28	99.10
C-2	99.71	99.40	99.78	99.27	99.08
C-3	99.66	99.60	99.68	99.15	98.94
C-4	99.77	99.10	99.93	99.41	99.27
C-5	99.66	99.37	99.73	99.15	98.94
Average	**99.70**	**99.25**	**99.81**	**99.25**	**99.07**
			Experiment-5		
C-1	99.31	98.17	99.60	98.28	97.85
C-2	99.30	98.40	99.52	98.25	97.82
C-3	99.34	98.20	99.62	98.35	97.94
C-4	99.45	98.90	99.58	98.62	98.28
C-5	99.29	98.07	99.60	98.23	97.79
Average	**99.34**	**98.35**	**99.59**	**98.35**	**97.93**

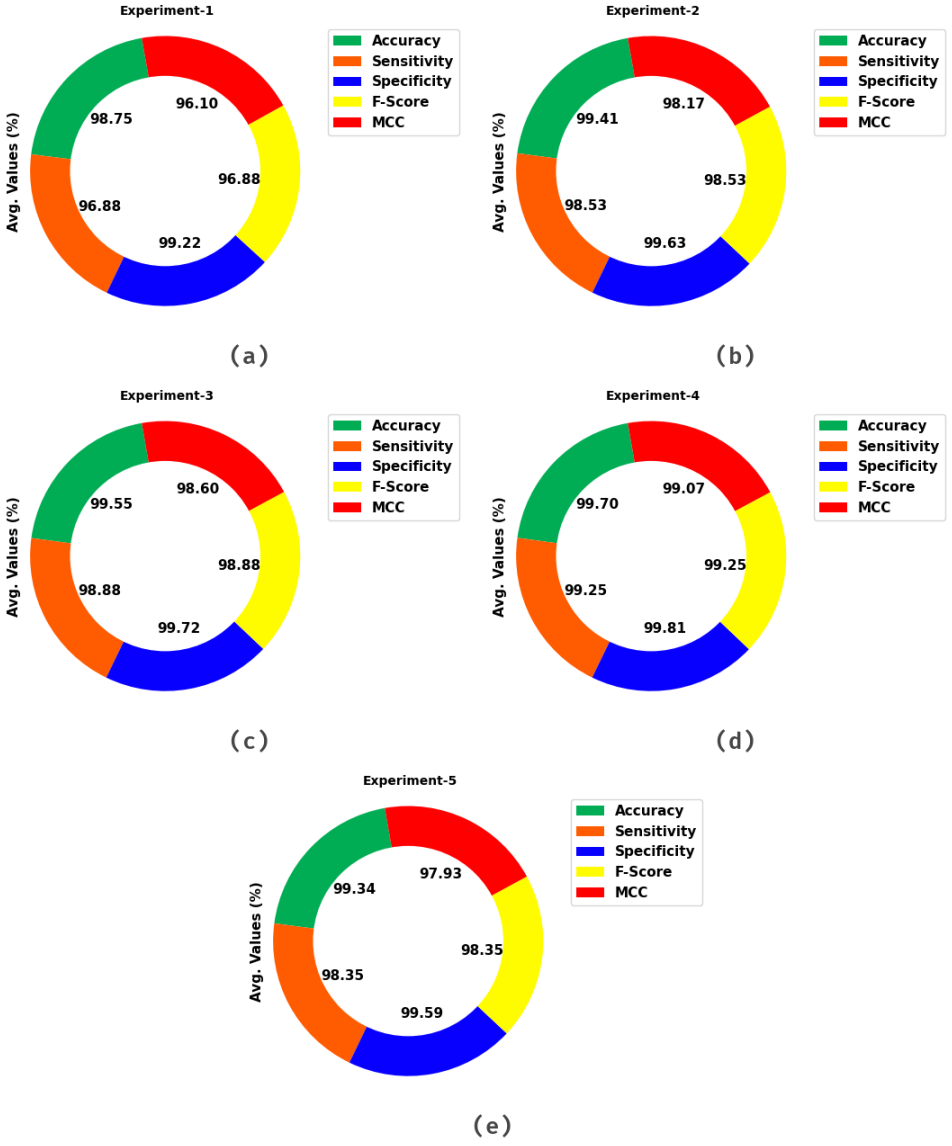

Figure 4. Average analysis of AOEDBC-DL system (**a**) Experiment-1, (**b**) Experiment-2, (**c**) Experiment-3, (**d**) Experiment-4, and (**e**) Experiment-5.

The training accuracy (TR_{acc}) and validation accuracy (VL_{acc}) acquired through the AOEDBC-DL system under the test database is exhibited in Figure 5. The simulation result stated that the AOEDBC-DL algorithm has realized increased values of TR_{acc} and VL_{acc}. Notably, the VL_{acc} appears to be superior to TR_{acc}.

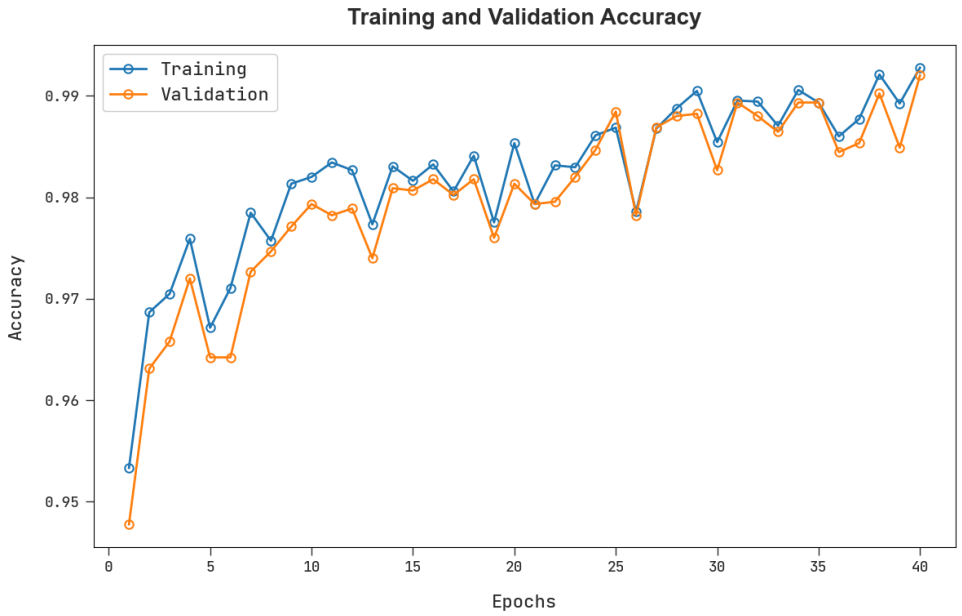

Figure 5. TR_{acc} and VL_{acc} analysis of AOEDBC-DL system.

The training loss (TR_{loss}) and validation loss (VL_{loss}) realized by the AOEDBC-DL methodology under the test database are displayed in Figure 6. The simulation result pointed out that the AOEDBC-DL system has achieved decreased values of TR_{loss} and VL_{loss}. The VL_{loss} is less than TR_{loss}.

Figure 6. TR_{loss} and VL_{loss} analysis of AOEDBC-DL system.

An apparent precision recall examination of the AOEDBC-DL system under the test database is showcased in Figure 7. The figure shows that the AOEDBC-DL methodology has resulted in higher values of precision recall value under different classes.

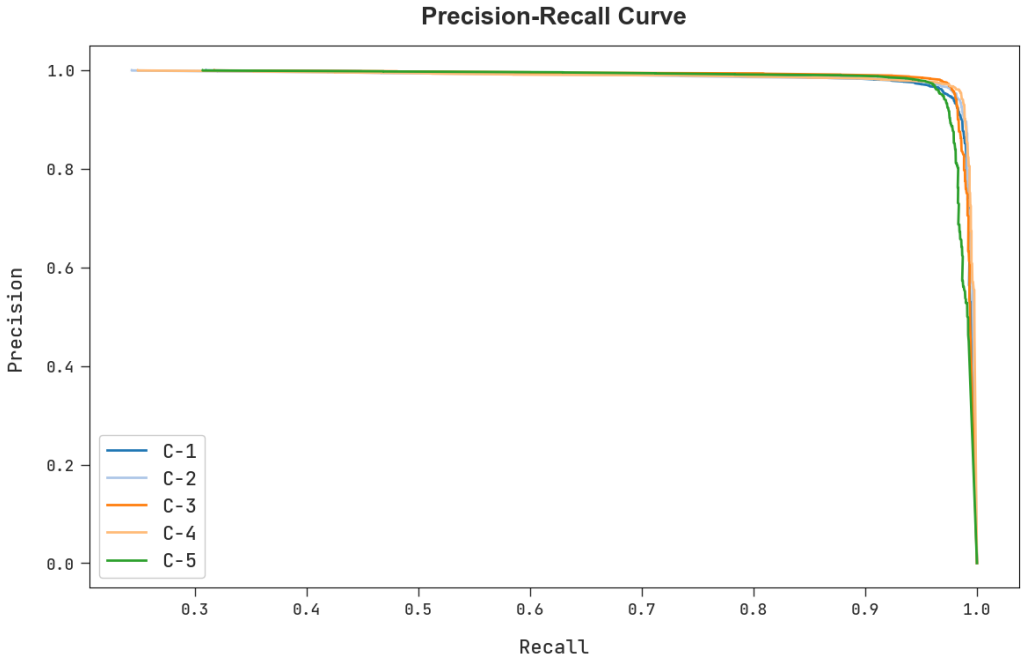

Figure 7. Precision recall analysis of AOEDBC-DL system.

A detailed ROC (receiver operator characteristic) analysis of the AOEDBC-DL approach under the test database is established in Figure 8. The outcomes referring to the AOEDBC-DL algorithm have exhibited their capability to classify varying classes.

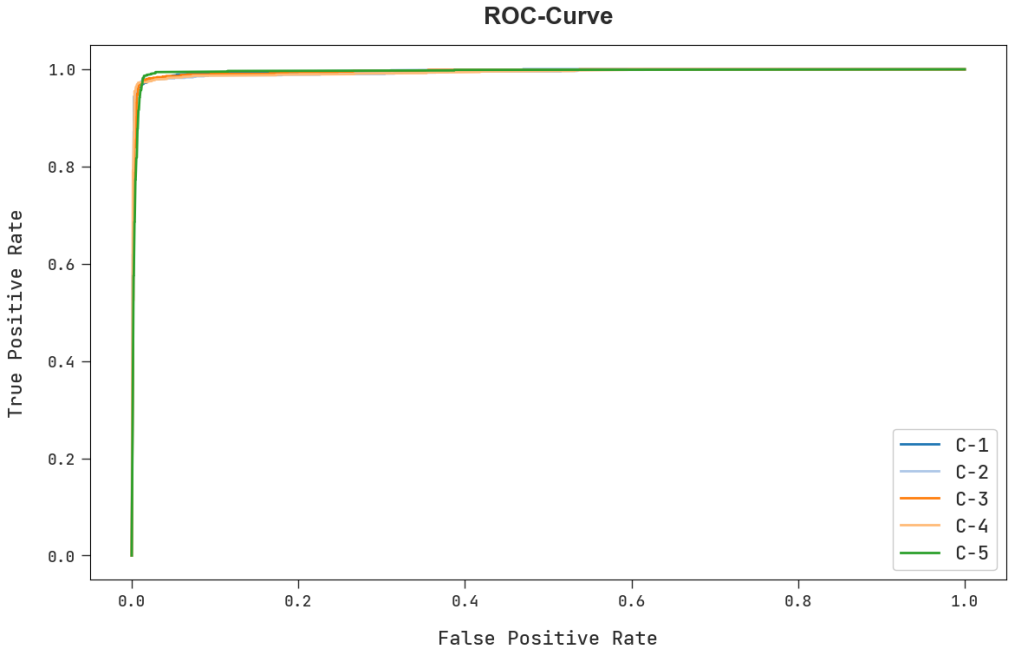

Figure 8. ROC curve analysis of AOEDBC-DL system.

To exhibit the superior performance of the AOEDBC-DL model, a detailed study analysis has been performed, shown in Table 3 and Figure 9 [28]. The experimental values imply that the AOEDBC-DL model has shown enhanced results over other models. Based on $accu_y$, a higher $accu_y$ of 99.70% is accomplished by the AOEDBC-DL model. In contrast, the KNN (K-Nearest Neighbor), KNN-PSO, KNN-AOA, AdaBoost, GB, and XGBoost models have shown decreased $accu_y$ values of 98.16%, 97.60%, 97.15%, 96.97%, 96.59%, and 95.41% correspondingly. Alternatively, with respect to F_{score}, the AOEDBC-DL model

has revealed superior F_{score} values of 98.54%, 95.98%, 96.27%, 95.81%, 98.28%, and 96.52% respectively. Thus, the AOEDBC-DL model has assured the enhanced intrusion detection outcomes of the AOEDBC-DL model.

Table 3. Comparative analysis of AOEDBC-DL system with existing algorithms [28].

Methods	Accuracy	Sensitivity	Specificity	F_{score}
AOEDBC-DL	99.70	99.25	99.81	99.25
KNN	98.16	98.32	97.01	98.54
KNN-PSO	97.60	98.62	96.58	95.98
KNN-AOA	97.15	96.30	98.64	96.27
AdaBoost	96.97	97.19	97.40	95.81
Gradient Boosting	96.59	97.96	95.60	98.28
XGBoost	95.41	96.05	98.47	96.52

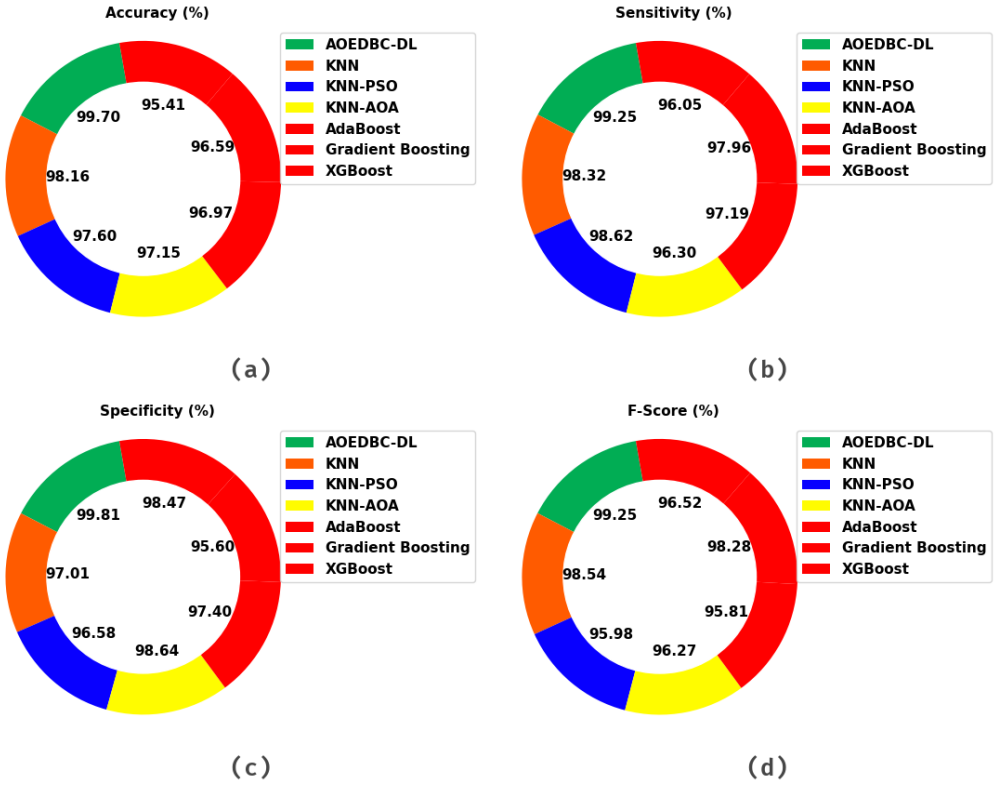

Figure 9. Comparative analysis of AOEDBC-DL system (**a**) $accu_y$, (**b**) $sens_y$, (**c**) $spec_y$, and (**d**) F_{score}.

5. Conclusions

In this work, we have developed a new AOEDBC-DL model for intelligent intrusion detection. The presented AOEDBC-DL technique first carried out the data clustering process to handle the massive quantity of network data traffic. The AOEDBC-DL technique applied a density-based clustering technique and the initial set of clusters were initialized using the AOA. To recognize intrusions, the QBA with the BiLSTM model was utilized in this study. The experimental result analysis of the AOEDBC-DL algorithm was tested using benchmark IDS datasets. Extensive comparison studies highlighted the enhancements of the AOEDBC-DL technique over other existing approaches. Thus, the presented AOEDBC-DL model can be applied to recognizing intrusions in the network. In the future, the performance of the AOEDBC-DL algorithm can be boosted by the feature selection and feature reduction processes.

Author Contributions: Conceptualization, R.M.; Data curation, R.M.; Funding acquisition, F.A.; Investigation, M.I.A.; Methodology, R.M. and A.M.H.; Project administration, A.M.H.; Resources, H.M. and M.I.A.; Software, H.M., I.Y. and G.P.M.; Supervision, F.A.; Validation, R.M., M.K.N., A.M.H. and I.Y.; Visualization, H.M. and I.Y.; Writing–original draft, F.A., M.K.N. and G.P.M.; Writing–review & editing, A.M.H., M.I.A. and G.P.M. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R77), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4310373DSR41).

Institutional Review Board Statement: This article does not contain any studies with human participants performed by any of the authors.

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 no conflict of interest.

References

1. Kalimuthan, C.; Renjit, A.J. Review on intrusion detection using feature selection with machine learning techniques. *Mater. Today Proc.* **2020**, *33*, 3794–3802. [CrossRef]
2. Taher, K.A.; Jisan, B.M.Y.; Rahman, M.M. Network intrusion detection using supervised machine learning technique with feature selection. In Proceedings of the International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019; pp. 643–646.
3. Prachi, H.M.; Sharma, P. Intrusion detection using machine learning and feature selection. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 43–52.
4. Wu, C.; Li, W. Enhancing intrusion detection with feature selection and neural network. *Int. J. Intell. Syst.* **2021**, *36*, 3087–3105. [CrossRef]
5. Anwer, H.M.; Farouk, M.; Abdel-Hamid, A. April. A framework for efficient network anomaly intrusion detection with features selection. In Proceedings of the 9th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 3–5 April 2018; pp. 157–162.
6. Alhakami, W.; ALharbi, A.; Bourouis, S.; Alroobaea, R.; Bouguila, N. Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. *IEEE Access* **2019**, *7*, 52181–52190. [CrossRef]
7. Pranto, M.B.; Ratul, M.H.A.; Rahman, M.M.; Diya, I.J.; Zahir, Z.B. Performance of machine learning techniques in anomaly detection with basic feature selection strategy-a network intrusion detection system. *J. Adv. Inf. Technol.* **2022**, *13*. [CrossRef]
8. Chen, H.; Miao, F.; Chen, Y.; Xiong, Y.; Chen, T. A hyperspectral image classification method using multifeature vectors and optimized KELM. *J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2781–2795. [CrossRef]
9. Wu, D.; Wu, C. Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows. *Agriculture* **2022**, *12*, 793. [CrossRef]
10. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Zhou, Y.; Chen, H.; Deng, W. Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization. *Knowl.-Based Syst.* **2021**, *224*, 107080. [CrossRef]
11. Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. *Appl. Sci.* **2022**, *12*, 5015. [CrossRef]
12. Al-Yaseen, W.L.; Idrees, A.K.; Almasoudy, F.H. Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognit.* **2022**, *132*, 108912. [CrossRef]
13. Li, X.; Chen, W.; Zhang, Q.; Wu, L. Building auto-encoder intrusion detection system based on random forest feature selection. *Comput. Secur.* **2020**, *95*, 101851. [CrossRef]
14. Gopalakrishnan, B.; Purusothaman, P. A new design of intrusion detection in IoT sector using optimal feature selection and high ranking-based ensemble learning model. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 2199–2226. [CrossRef]
15. Faysal, J.A.; Mostafa, S.T.; Tamanna, J.S.; Mumenin, K.M.; Arifin, M.M.; Awal, M.A.; Shome, A.; Mostafa, S.S. January. XGB-RF: A hybrid machine learning approach for IoT intrusion detection. *Telecom* **2022**, *3*, 52–69. [CrossRef]
16. 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]
17. Çavuşoğlu, Ü. A new hybrid approach for intrusion detection using machine learning methods. *Appl. Intell.* **2019**, *49*, 2735–2761. [CrossRef]
18. Li, S.S. An improved DBSCAN algorithm based on the neighbor similarity and fast nearest neighbor query. *IEEE Access* **2020**, *8*, 47468–47476. [CrossRef]

19. Ullah, S.; Khan, M.A.; Ahmad, J.; Jamal, S.S.; e Huma, Z.; Hassan, M.T.; Pitropakis, N.; Buchanan, W.J. HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. *Sensors* **2022**, *22*, 1340. [CrossRef]
20. Otoum, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning-based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3803.
21. Mendonça, R.V.; Silva, J.C.; Rosa, R.L.; Saadi, M.; Rodriguez, D.Z.; Farouk, A. A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. *Expert Syst.* **2022**, *39*, e12917. [CrossRef]
22. Nasir, M.; Javed, A.R.; Tariq, M.A.; Asim, M.; Baker, T. Feature engineering and deep learning-based intrusion detection framework for securing edge IoT. *J. Supercomput.* **2022**, *78*, 8852–8866. [CrossRef]
23. Khatir, S.; Tiachacht, S.; Le Thanh, C.; Ghandourah, E.; Mirjalili, S.; Wahab, M.A. An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates. *Compos. Struct.* **2021**, *273*, 114287. [CrossRef]
24. Liu, X.; Liu, S.; Li, X.; Zhang, B.; Yue, C.; Liang, S.Y. Intelligent tool wear monitoring based on parallel residual and stacked bidirectional long short-term memory network. *J. Manuf. Syst.* **2021**, *60*, 608–619. [CrossRef]
25. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Cham, Switzerland, 2010; pp. 65–74.
26. Islam, J.; Nazir, A.; Hossain, M.M.; Alhitmi, H.K.; Kabir, M.A.; Jallad, A.H.M. A Surrogate Assisted Quantum-behaved Algorithm for Well Placement Optimization. *IEEE Access* **2022**, *10*, 17828–17844. [CrossRef]
27. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. *J. Sens.* **2016**, *2016*, 4731953. [CrossRef]
28. Liu, G.; Zhao, H.; Fan, F.; Liu, G.; Xu, Q.; Nazir, S. An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. *Sensors* **2022**, *22*, 1407. [CrossRef]