electronics

MDPI

Article

Optimized Machine Learning-Based Intrusion Detection System for Fog and Edge Computing Environment

Omar A. Alzubi [1,*], Jafar A. Alzubi [2,*], Moutaz Alazab [3,*], Adnan Alrabea [1], Albara Awajan [3] and Issa Qiqieh [2]

[1] Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt 19117, Jordan
[2] Faculty of Engineering, Al-Balqa Applied University, Al-Salt 19117, Jordan
[3] Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt 19117, Jordan
* Correspondence: o.alzubi@bau.edu.jo (O.A.A.); j.alzubi@bau.edu.jo (J.A.A.); m.alazab@bau.edu.jo (M.A.)

Abstract: As a new paradigm, fog computing (FC) has several characteristics that set it apart from the cloud computing (CC) environment. Fog nodes and edge computing (EC) hosts have limited resources, exposing them to cyberattacks while processing large streams and sending them directly to the cloud. Intrusion detection systems (IDS) can be used to protect against cyberattacks in FC and EC environments, while the large-dimensional features in networking data make processing the massive amount of data difficult, causing lower intrusion detection efficiency. Feature selection is typically used to alleviate the curse of dimensionality and has no discernible effect on classification outcomes. This is the first study to present an Effective Seeker Optimization model in conjunction with a Machine Learning-Enabled Intrusion Detection System (ESOML-IDS) model for the FC and EC environments. The ESOML-IDS model primarily designs a new ESO-based feature selection (FS) approach to choose an optimal subset of features to identify the occurrence of intrusions in the FC and EC environment. We also applied a comprehensive learning particle swarm optimization (CLPSO) with Denoising Autoencoder (DAE) for the detection of intrusions. The development of the ESO algorithm for feature subset selection and the DAE algorithm for parameter optimization results in improved detection efficiency and effectiveness. The experimental results demonstrated the improved outcomes of the ESOML-IDS model over recent approaches.

Keywords: security; machine learning; fog computing; intrusion detection system; optimization; feature selection; edge computing

check for updates

Citation: Alzubi, O.A.; Alzubi, J.A.; Alazab, M.; Alrabea, A.; Awajan, A.; Qiqieh, I. Optimized Machine Learning-Based Intrusion Detection System for Fog and Edge Computing Environment. *Electronics* **2022**, *11*, 3007. https://doi.org/10.3390/electronics11193007

Academic Editors: Celestine Iwendi and Thippa Reddy Gadekallu

Received: 25 August 2022
Accepted: 16 September 2022
Published: 22 September 2022

1. Introduction

Due to the tremendous growth of smart devices, we are approaching the era of the Internet of Things (IoT) [1]. The IoT application requires geo-distribution, mobility support, low latency, and location awareness, all of which are difficult to implement in cloud computing (CC). Edge paradigms such as mobile edge computing (MEC) and fog computing (FC) are presented to overcome IoT implementation challenges [2–4]. The nodes, which can implement computing tasks, are named MEC hosts or fog nodes in MEC and FC hosts in MEC, which could offer lower-latency services.

MEC and FC are slightly different; MEC hosts are typically installed by mobile service providers, whereas FC is composed of an edge server or device with computing and communication power [5], and FC is composed of an edge server or device with computing and communication power. Researchers extended CC to the edge by analogizing the network system and various characteristics [6]. A new network model, such as MEC or FC, provokes several concerns regarding network performance and stability. Figure 1 depicts the structure of fog edge computing that is employed in MEC and FC.

The majority of terminal devices in MEC or FC are resource-constrained; the terminal connected to the MEC or FC hosts could be an unmanned aerial vehicle (UAV), a smart home appliance, a VR device, or a smartphone [7]. Man in the middle (MIM), denial of service (DoS), privacy leakage, service manipulation, and rogue gateway are all possible attacks on the FC system [8,9]. We focused on privacy protection as an effective method for detecting the presence of intruders, assisting in the combatting of security threats in FC structures, and reducing the resulting cybersecurity damages.

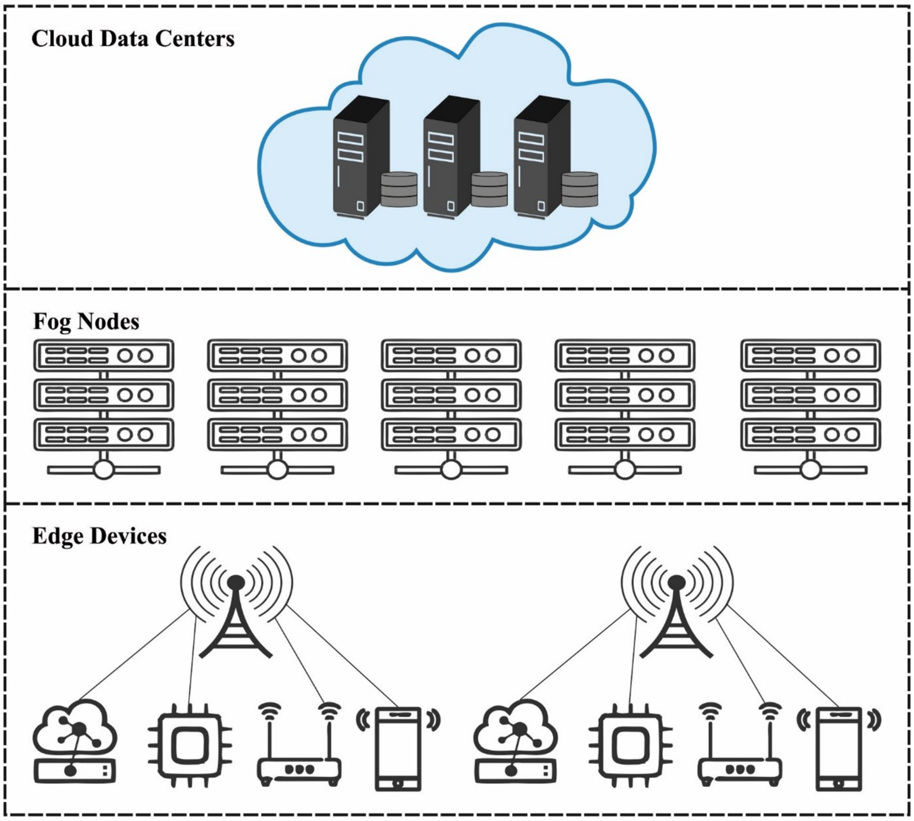

Figure 1. Fog edge computing.

The intrusion detection system for fog and edge computing environments detects intruders in two ways: anomaly-based detection and signature-based detection, The normal behavior of the scheme is taken into account as a model in anomaly-based detection, which then examines the behavior of incoming traffic and categorizes it as either normal or abnormal based on the model that was built [10–12]. In contrast, signature-based detection compares incoming traffic to pre-established rules to determine whether to allow or reject it. In the past few years, there have been a variety of study articles developed in the area of intrusion detection systems for fog and edge computing environments [13,14]. Early research concentrated on supervised machine learning and unsupervised machine learning. There have also been attempts to implement advanced applications [15–17], such as a conventional detection method that allows the incorporation of the results of various classifications to effectively improve IDS performance.

This study introduces an Effective Seeker Optimization with Machine Learning-Enabled Intrusion Detection System (ESOML-IDS) model for FC and EC environments. The ESOML-IDS model intends to appropriately determine the existence of intrusions in the FC and EC environment. The ESOML-IDS model derives a novel ESO-based feature selection (FS) approach to choose an optimal subset of features. Moreover, comprehensive learning particle swarm optimization (CLPSO) with Denoising Autoencoder (DAE) is applied for the detection and classification of intrusions. In order to demonstrate the enhanced outcomes of the ESOML-IDS model, a wide range of simulations was carried out.

Contributions of This Study

The main contributions of this study are as follows:

- We develop a new Effective Seeker Optimization with Machine Learning-Enabled Intrusion Detection System (ESOML-IDS) technique for intrusion detection and classification in FC and EC environments;
- To detect and classify intrusions, a group of sub-processes are incorporated with the proposed technique, including pre-processing, ESO-based feature subset selection, a DAE classifier, and CLPSO-based parameter optimization;
- To demonstrate the comparative advantages of the proposed technique over recent approaches, a wide variety of exhaustive simulations are carried out.

The rest of the paper is organized as follows. Section 2 surveys relevant research in this area of intrusion detection. Section 3 introduces the proposed model. Section 4 provides performance validation, showing the comparative advantages of applying the proposed techniques in terms of cost, accuracy, and comparative analysis. Finally, Section 5 concludes the paper.

2. Related Works

Lin et al. [18] presented a resource allocation and IDS architecture in edge computing. In particular, the presented method is developed to aid heterogeneous resource-demanding allocation and resource sharing. An edge computing IDS is introduced, and utilizing this approach is the foundation for resource allocation. Next, a single-layer dominant and max-min fair (SDMMF) allocation was employed. Li et al. [19] employed the game concept in the field of edge computing systems and recommended a data-driven mimicry ID game theory-based named GLIDE. The game income of participants and the utility computation method under distinct positioning approaches were analyzed. Wang et al. [20] presented an architecture for optimizing the smart false alarm reduction for DIDS-based edge computing devices. The proposed method could offer energy efficacy as the data could be treated at the edge for a short response time. The assessment result demonstrated that the architecture could assist in reducing the task for the central server and the delay in comparison with the comparative study.

Sudqi Khater et al. [21] presented a lightweight IDS-based vector space depiction with an MLP method. Next, they estimated the proposed method against the Australian Defense Force Academy Windows Dataset (ADFA-WD) and ADFA with Linux Dataset (ADFA-LD), which is a novel generation system dataset that comprises exploits and attacks on different applications. An et al. [22] presented a hypergraph clustering method based on the Apriori approach. Our study could efficiently determine the relationship among FC that is suffering from the threats of DDoS. Next, they verified that the resource consumption rate of the model could be efficiently promoted via DDoS analysis.

Mourad et al. [23] developed a vehicular edge computing (VEC) fog-assisted system that allows the offloading of IDS tasks to federated vehicle nodes situated within the Adhoc vehicular fog that is implemented with minimum latency. Abdel-Basset et al. [24] introduced a forensics-based DL (Deep-IFS) for identifying intrusions in IIoT traffic. The presented approach learns local representations with LocalGRU and presents an MHA to learn and capture global representations (with longer-range dependency). A residual connection among layers is developed for preventing data loss. Pacheco et al. [25] proposed an Anomaly Behavior Analysis Method based on ANN, to obtain an adaptive IDS that could be able to detect whether a fog node was compromised and also take proper action for ensuring transmission accessibility.

3. The Proposed Model

In this study, a novel ESOML-IDS approach was developed for intrusion detection and classification in FC and EC environments. The presented ESOML-IDS technique aimed to identify the occurrence of intrusions in the FC and EC environment. The ESOML-IDS technique encompasses a series of sub-processes such as pre-processing, ESO-based feature subset selection, a DAE classifier, and CLPSO-based parameter optimization.

3.1. Data Normalization

The *z*-score is a conventional standardized and normalized approach that signifies the number of standard deviations (SD). It normalizes the data set to the above-mentioned scale for converting every datum with a distinct scale to the default scale.

For normalizing the data utilizing the *z*-score, it can be subtracted the mean of populations in a rare data point and separated by the SD that offers a score ideally different amongst −3 and +3, thus reflecting that a point is several SDs above/below the mean, as calculated by Equation (1), where x signifies the value of the specific sample, μ stands for the mean and σ denotes the SD.

$$z\_score = \frac{x - \mu}{\sigma} \tag{1}$$

3.2. Design of ESO-Based Feature Selection Technique

The elastic collision seeker optimization algorithm (ESOA) involved in [26] has been employed in our system for feature selection. The seeker optimization algorithm (SOA) implements an in-depth search simulating human search performance. The SOA is optimized as a search for the most optimal solution with a team of explorers in exploring space, using the search team as the population and the seeker as the task approach. Three significant upgrading stages are called ESOs.

3.2.1. Search Direction

The forward orientation of searching is determined as the experience gradient attained in the individual effort and the estimation of another individual searching a past place. The egoistic path $\overrightarrow{f}_{i.e}(t)$, altruistic path $\overrightarrow{f}_{i.a}(t)$, and preemptive path $\overrightarrow{f}_{i.p}(t)$ of ith individual from some dimension are achieved.

$$\begin{aligned} \overrightarrow{f}_{i.e}(t) &= \overrightarrow{p}_{i,best} - \overrightarrow{x}_{i}(t) \\ \overrightarrow{f}_{i.a}(t) &= \overrightarrow{g}_{i,best} - \overrightarrow{x}_{i}(t) \\ \overrightarrow{f}_{i.p}(t) &= \overrightarrow{x}_{t1} - \overrightarrow{x}_{t2} \end{aligned} \tag{2}$$

The seeker utilizes the technique of an arbitrary weighted average for obtaining the search orientation.

$$\overrightarrow{f}_{i}(t) = sign(\omega \overrightarrow{f}_{i.p}(t) + \phi_1 \overrightarrow{f}_{i.e}(t) + \phi_2 \overrightarrow{f}_{i.a}(t)) \tag{3}$$

where $t_1, t_2 \in \{t, t-1, t-2\}$; $\overrightarrow{x}_{i}(t_1)$ and $\overrightarrow{x}_{i}(t_2)$ are the optimum benefits of $\overrightarrow{x}_{i}(t-2)$, $\overrightarrow{x}_{i}(t-1)$, $\overrightarrow{x}_{i}(t)$ individually; $g_{i,best}$ refers to the historical optimum place from the neighborhood, where the ith searching factor was placed; $p_{i,best}$ represents the optimum locality in the ith searching factor to present locality; ψ_1 is an arbitrary number from zero to one, and ω implies the weight of inertia.

3.2.2. Search Step Size

The ESO represents the capability of fuzzy approximation reasoning. The technique adjusts to the best estimate of the objectively optimized problem when it expresses a simple fuzzy rule. Greater significance is associated with longer searching stages, whereas lower fitness corresponds to shorter searching stages. The Gaussian distribution function was adapted for describing the search step measurements.

$$\mu(a) = e^{\frac{\alpha^2}{2\delta^2}} \tag{4}$$

where α and δ represent the parameters of membership functions. Based on Equation (4), the probability of a resultant variable above $[-3\delta, 3\delta]$ is less than 0.0111. Thus, $\mu_{min} = 0.0111$. However, for accelerating the convergence speed and attaining an optimum individual to take an undefined step size, μ_{max} is fixed as 0.9.

$$\mu(i) = \mu_{max} - \frac{s - I_i}{s - I}(\mu_{max} - \mu_{min}), i = 1, 2, \ldots, s \tag{5}$$

$$\mu ij = rand(\mu_i), j = 1, 2, \ldots, D \tag{6}$$

where μ_{ij} has been defined in Equations (5) and (6), I_i refers to the number of sequences $X(t)$ of the current individuals set in higher to lower function values, and the function refers to the real number from some partition $[\mu_i, 1]$. It is realized that Equation (5) reflects the arbitrary search performance of human beings. The step measurement of j dimension searching the interspace is defined in the subsequent formula:

$$\alpha_{ij} = \delta_{ij} - \sqrt{-\ln(\mu_{ij})} \tag{7}$$

where δ_{ij} refers to the parameter of the Gaussian distribution function that is demonstrated in Equations (8) and (9):

$$\omega = \frac{iter_{max} - t}{iter_{max}} \tag{8}$$

$$\delta_{ij} = \omega * abs(\overrightarrow{x}_{min} - \overrightarrow{x}_{max}) \tag{9}$$

where ω refers to the weight of inertia. While the evolutionary algebra improves, ω reduces linearly from 0.9 to 0.1. $\overrightarrow{x}_{min}$ and $\overrightarrow{x}_{max}$, correspondingly, denote the variates of the minimal and maximal values of the function. Figure 2 depicts the flowchart of SOA.

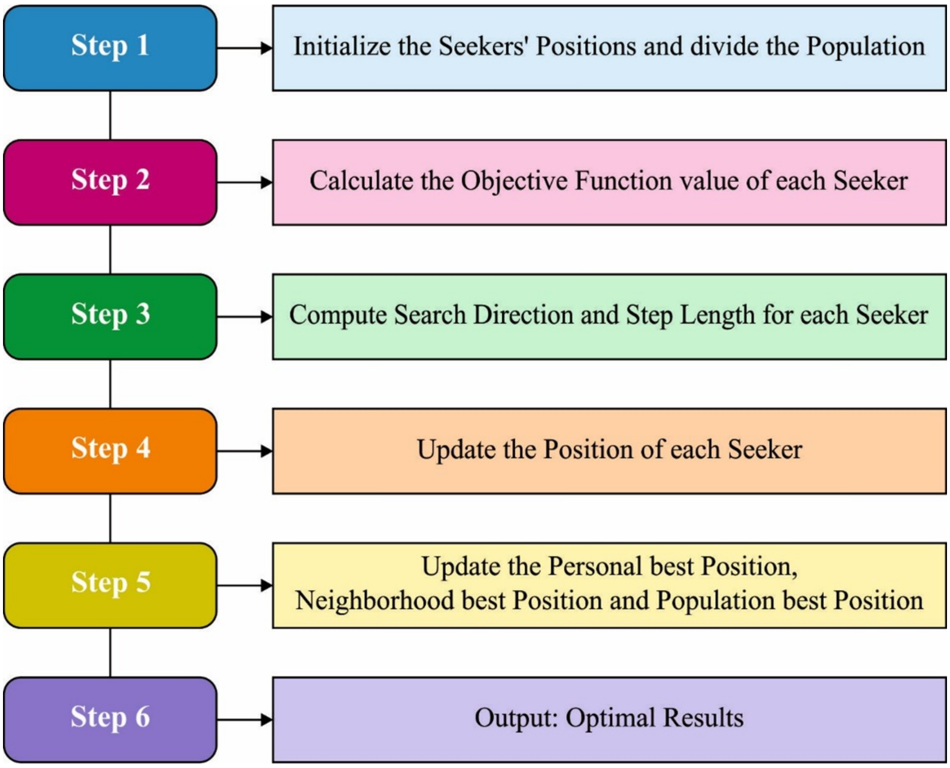
Step 1
Initialize the Seekers' Positions and divide the Population
Step 2
Calculate the Objective Function value of each Seeker
Step 3
Compute Search Direction and Step Length for each Seeker
Step 4
Update the Position of each Seeker
Step 5
Update the Personal best Position,
Neighborhood best Position and Population best Position
Step 6
Output: Optimal Results

Figure 2. Flowchart of SOA.

3.2.3. Individual Location Updates

After obtaining the scout path and scout step measurement of individuals, the place upgrade is expressed as in Equation (10):

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) f_{ij}(t), i = 1, 2, \ldots s; j = 1, 2, \ldots, D \tag{10}$$

i refers to the ith searching individual; j signifies the individual dimensional; $f_{ij}(t)$ and $\alpha_{ij}(t)$, correspondingly, represent the seekers' path and searching step size at time t; and $x_{ij}(t)$ and $x_{ij}(t+1)$, correspondingly, define the seekers' site at time t and $(t+1)$.

The mathematical model of the ESO-FS approach was established. Usually, the classification (for instance, supervised learning) requires some datasets that are of size $N_S \times N_F$, whereas N_S signifies the count of samples and N_F defines the count of features. The main function of the FS problem is selecting a subset of features S in the entire amount of features (N_F), whereas the size of S is less than N_F. It is attained by minimizing the subsequent main function:

$$Fit = \lambda \times \gamma_s + (1 - \lambda) \times (\frac{|S|}{N_F}) \tag{11}$$

where γ_s implies the classifier error utilizing S and $|S|$ as the count of chosen features. λ is utilized for balancing amongst $(\frac{|S|}{N_F})$ and γ_s.

3.3. Process Involved in DAE-Based Classification

During the intrusion detection process, the chosen features are fed into the DAE model to classify intrusions [27]. DAE is dependent upon the AE. Noise (Gaussian noise usually, or setting the data to zero arbitrarily) is present in trained data, and AE is required to be learned for removing noise so as to obtain uncontaminated input data. In the case of corrupted input, the AE is defined further as a stable and suitable feature that establishes a further advanced description of the input data and improves the robustness of the total method. At this point, x is the primary input data, x_1 is the corrupted input data, y is the novel feature attained by the encoded x_1, and z represents the outcome attained by the decoded y. The reconstructing error is calculated by Equation (12):

$$L_D = ||x - g(f(x_1))||^2 \tag{12}$$

The cost function is computed as:

$$JD(W, b) = \left[\frac{1}{m} \sum_{i=1}^{N} (\frac{1}{2} ||x^i - g(f(x_1^i))||^2\right] + \frac{\lambda}{2} \sum_{l=1}^{2} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^l)^2) \tag{13}$$

Generally, it is only required to arbitrarily fix the unit from x to zero based on the noise figure k ($k \in [0, 1]$); afterward, x_1 is attained. This technique of resolving the parameters is similar to that of AE. Figure 3 displays the infrastructure of DAE.

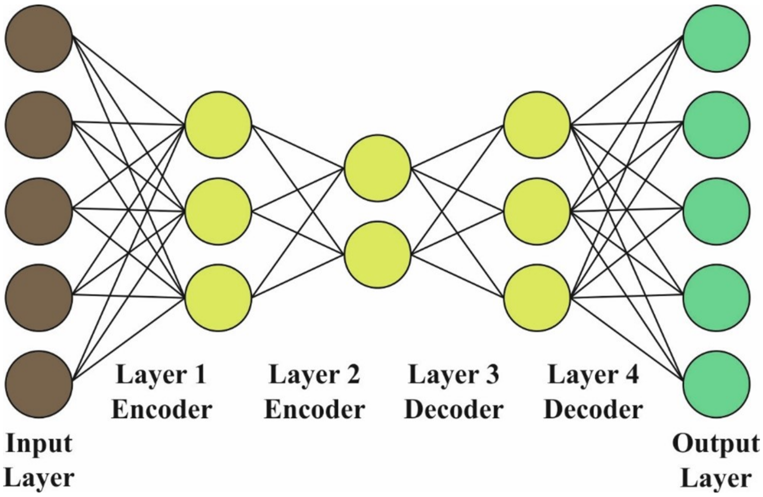

Figure 3. DAE structure.

3.4. Parameter Tuning Using CLPSO Algorithm

We used the CLPSO algorithm, developed by [28], to achieve optimal tuning of the parameters involved in the DAE model. PSO is a typical evolutionary computing approach stimulated in the analysis of the predation performance of birds; the basic concept of the PSO technique is sharing cooperation and data amongst individuals for finding the optimum solutions. The velocity signifies the speed and direction in which the particle moves. The position signifies the particle's position. In order to process all the particles, only the individual optimum experience and the global optimum experience of the total swarm are learned. Assume $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$, which refer to the position and velocity of particle i $i = \{1, 2, \ldots, N\}$, correspondingly, whereas D refers to the dimensions of the primary space and N represents the population size. Assume that $pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})^T$ and $gbest = (gbest_1, gbest_2, \ldots, gbest_D)^T$ exist as the individual optimum place of particles i and the global optimum position of the entire swarm. The upgrade of velocity, as well as the position of particles, is computed by Equations (14) and (15):

$$v_{id} = w * v_{id} + c_1 * rand_1(0,1) * (pbest_{id} - x_{id}) + c_2 * rand_2(0,1) * (gbest_{id} - x_{id}) \tag{14}$$

$$x_{id} = x_{id} + v_{id} \tag{15}$$

where $i = 1, 2, \ldots, N$ and $d = 1, 2, \ldots, D$. However, w refers to the inertia weight, c_1 and c_2 stand for the acceleration co-efficient, and $rand_1(0,1)$ and $rand_2(0,1)$ are uniform arbitrary numbers.

The CLPSO algorithm adapts the approach of comprehensive learning for selecting an object for learning, rather than learning by themselves, and the global optimum individual [28]. The velocity upgrading formula in CLPSO is determined as:

$$v_{id} = w * v_{id} + c_1 * rand_1(0,1) * (pbest_{f_i d} - x_{id}) \tag{16}$$

where f_i determines that particle *pbests* is the particle that i must follow, and $rand(0,1) \in [0,1]$ refers to a uniform arbitrary number. The CLPSO allocates the learning probability Pc_i to all the particles i utilizing the subsequent formula:

$$Pc_i = 0.05 + 0.45 \frac{exp(10(i-1)/N-1) - 1}{exp(10) - 1} \tag{17}$$

In order to obtain all the solutions x_i, it is learned from several particles rather than only two particles. All the components of particles i learn by themselves or by another particle depending upon learning probability Pc_i. Arbitrary components of particles i will learn from another particle when all their elements learn by themselves. The superior fitness value of a solution is the superior possibility in which a particle is learned.

The CPSO technique is used for determining *FF* with the objective of minimizing the classifier error rate, as provided below. The solution with the minimum classifier error rate is assumed as a better solution.

$$fitness(x_i) = Classifier\ Error\ Rate(x_i) = \frac{Number\ of\ misclassified\ instances}{Total\ number\ of\ instances} * 100 \quad (18)$$

4. Empirical Results and Validation

This section discusses the effectiveness of applying the ECSOML-IDS technique to detect and classify intrusions under several varieties of FS methods and class labels. It demonstrates and validates the enhanced outcomes of employing the ECSOML-IDS technique in terms of a wide set of accuracy metrics. Thus, the experimental work of this manuscript, together with the cost and performance analysis, is described below.

4.1. Cost Analysis

The UNSW-NB15 datasets are used for experimental validation because they have significant potential for attack pattern recognition and analysis, as well as being effective in enhancing the effectiveness of intrusion classifiers. In contrast to NSL-KDD and KDD-CUP'99, Zhang et al. [28] claim that the UNSW-NB15 dataset better simulates the current network traffic environment; the dataset holds a set of 42 features, including 3 categorical and 39 numeric features. Table 1 and Figure 4 report the FS outcomes of the ESO-FS and other FS techniques in terms of the number of features chosen and best cost (BC).

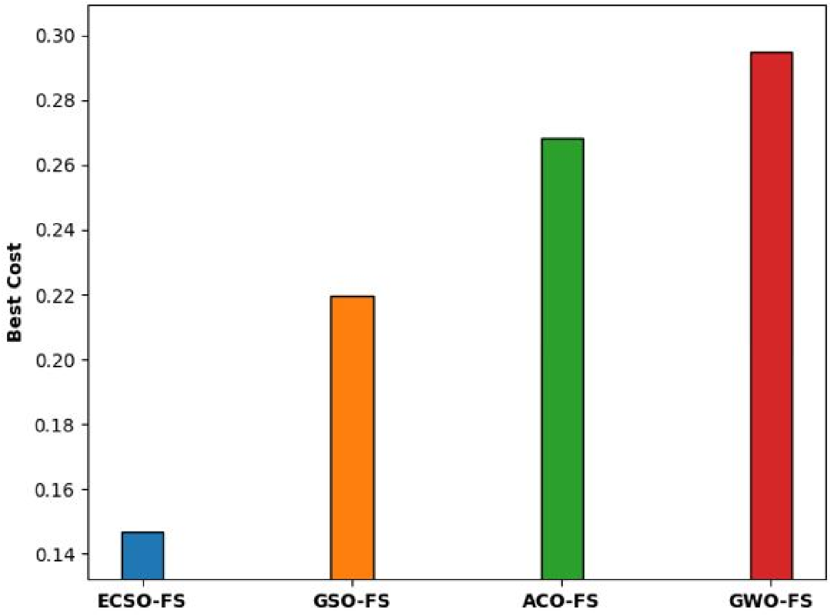

Figure 4. Best cost analysis of ESO-FS technique.

The results indicated that the GWO-FS model showcased worse FS outcomes with a BC of 0.947, whereas the ACO-FS technique obtained a slightly enhanced BC of 0.268. At the same time, the GSO-FS technique has resulted in a reasonable BC of 0.2194. However, the ESO-FS technique has displayed enhanced FS outcomes with the choice of 12 features and a BC of 0.1468.

Table 1. FS analysis of ESO-FS technique.

Methods	No. of Features Selected	Best Cost
ESO-FS	12	0.1468
GSO-FS	16	0.2194
ACO-FS	18	0.2681
GWO-FS	24	0.2947

4.2. Performance Measures and Analysis

In this subsection, the impact of accuracy derived from utilizing the ECSOML-IDS technique, for different numbers of epochs, and label classes, is examined. Several performance metrics have been discussed in [*] for evaluating the effectiveness and quantifying errors resulting from using certain class types with a distinct number of epochs. In this paper, for performance validation purposes, several accuracy metrics have been used, such as training accuracy, validation accuracy, testing accuracy, precision, recall, and F1 score, which are denoted by tr_accu, val_accu, $test_accu$, $prec_n$, $reca_l$, and F_score, respectively. Generally, classification accuracy is the ratio of the number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ numberof\ predictions\ made} \tag{19}$$

Moreover, the precision metric reflects the proportion of positive identifications that was actually correct. Therefore, precision is computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

Meanwhile, the recall is the fraction of relevant instances that were retrieved. The recall can be mathematically defined as:

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

where TP, FP, and FN are True Positive, False Positive, and False Negative outcomes, respectively. Moreover, the *F1-score* is the traditional F-measure or balanced F score *F1-score*) and is defined as the harmonic mean of precision and obtained as:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{TP}{TP + 1/2(FP + FN)} \tag{22}$$

Table 2 and Figure 5 portray the classification outcomes of the ESOML-IDS model under 1000 epochs and distinct classes. The results indicated that the ESOML-IDS model resulted in effective outcomes under every class. For instance, with a normal class, the ESOML-IDS model has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 83.38%, 83.56%, 78.22%, 82.72%, 81.50%, and 80.59% respectively. At the same time, with the DoS class, the ESOML-IDS model has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 83.14%, 83.50%, 80.18%, 82.10%, 83.47%, and 81.99%, correspondingly. Moreover, with the generic class, the ESOML-IDS system has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 82.46%, 82.88%, 80.43%, 82.08%, 81.21%, and 80.87%, correspondingly.

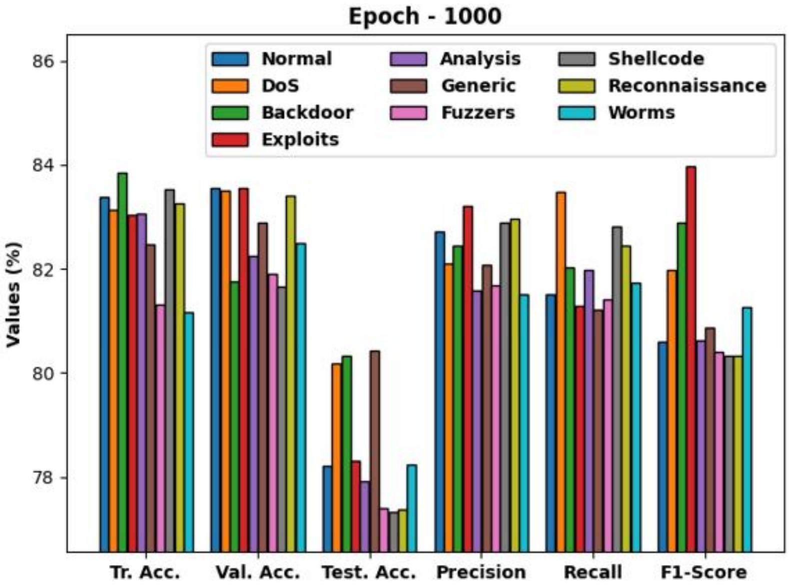

Figure 5. Result analysis of ESOML-IDS technique under 1000 epochs and distinct classes.

Table 2. Result analysis of ESOML-IDS technique under 1000 epochs and distinct classes.

Epoch-1000						
Class Labels	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**	**Precision**	**Recall**	**F1-Score**
Normal	83.38	83.56	78.22	82.72	81.50	80.59
DoS	83.14	83.50	80.18	82.10	83.47	81.99
Backdoor	83.86	81.75	80.32	82.45	82.02	82.89
Exploits	83.03	83.56	78.32	83.21	81.28	83.98
Analysis	83.07	82.24	77.92	81.59	81.99	80.63
Generic	282.46	82.88	80.43	82.08	81.21	80.87
Fuzzers	81.32	81.90	77.41	81.68	81.42	80.40
Shellcode	83.52	81.65	77.33	82.88	82.82	80.34
Reconnaissance	83.25	83.41	77.38	82.97	82.44	80.34
Worms	81.17	82.49	78.24	81.51	81.74	81.26
Average	**82.82**	**82.69**	**78.58**	**82.32**	**81.99**	**81.33**

Furthermore, with the shellcode class, the ESOML-IDS method has obtained tr_{accu},val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 83.52%, 81.65%, 77.33%, 82.88%, 82.82%, and 80.34%, respectively. Eventually, with the worms class, the ESOML-IDS approach has obtained tr_{accu},val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 81.17%, 82.49%, 78.24%, 81.51%, 81.74%, and 81.26%, correspondingly.

The accuracy outcome analysis of the ESOML-IDS approach on test data is exhibited in Figure 6. The results demonstrated that the ESOML-IDS technique achieved improved validation accuracy related to training accuracy. It is also observable that the accuracy values become saturated with the epoch count of 1000.

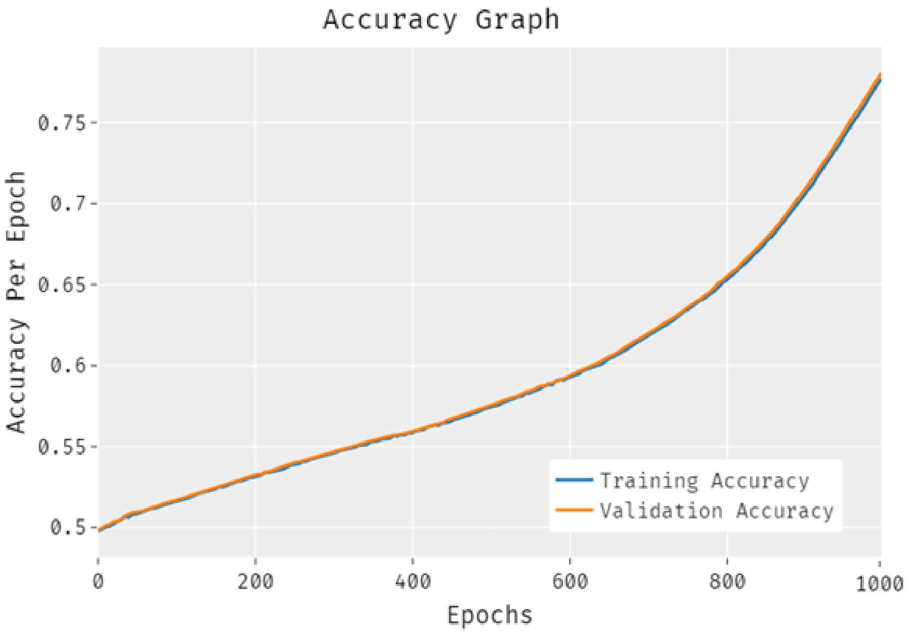

Figure 6. Accuracy analysis of ESOML-IDS technique under 1000 epochs.

The loss outcome analysis of the ESOML-IDS system on test data is demonstrated in Figure 7. The figure shows that the ESOML-IDS technique offers reduced validation loss in terms of training loss. It is additionally noticed that the loss values become saturated with an epoch count of 1000.

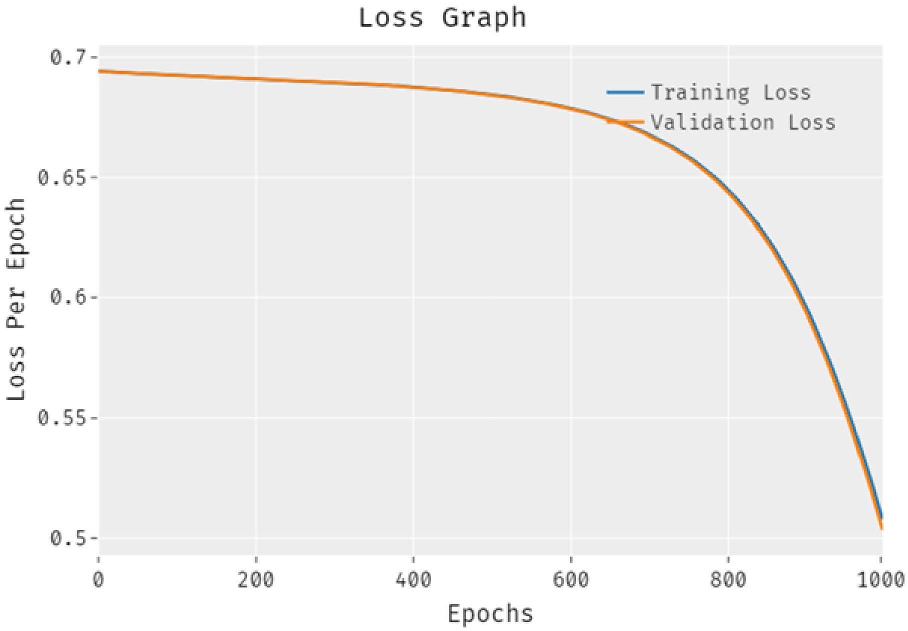

Figure 7. Loss analysis of ESOML-IDS technique under 1000 epochs.

Table 3 and Figure 8 portray the classification outcomes of the ESOML-IDS algorithm under 2000 epochs and distinct classes. The results indicated that the ESOML-IDS model resulted in effective outcomes under every class. For example, with the normal class, the ESOML-IDS technique has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 81.54%, 82.64%, 83.02%, 81.92%, 83.49%, and 82.18%, correspondingly. Simultaneously, with the DoS class, the ESOML-IDS approach has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 82.72%, 83.10%, 84.78%, 83.08%, 81.46%, and 83.82%, respectively. Moreover, with the generic class, the ESOML-IDS methodology has obtained tr_{accu}, val_{accu}, $test_{accu}$,

$prec_n$, $reca_l$, and F_{score} of 83.80%, 81.09%, 83.92%, 82.29%, 83.19%, and 83.97%, respectively. Moreover, with the shellcode class, the ESOML-IDS model has obtained tr_{accu},val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 83.42%, 81.37%, 83.29%, 83.35%, 82.15%, and 82.55%, correspondingly. At last, with the worms class, the ESOML-IDS model has obtained tr_{accu}, val_{accu}, $test_{accu}$, $prec_n$, $reca_l$, and F_{score} of 82.64%, 83.59%, 82.82%, 82.74%, 82.68%, and 83.75%, correspondingly.

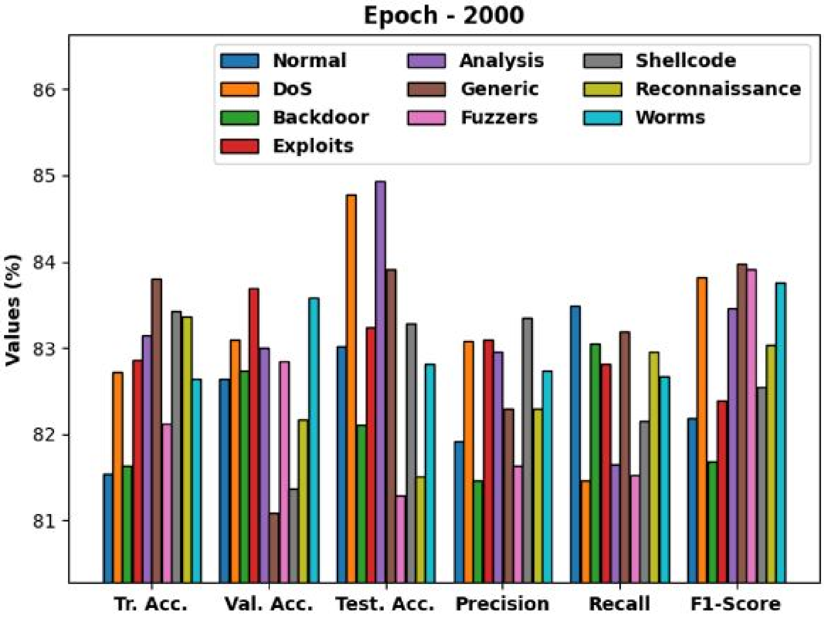

Figure 8. Result analysis of ESOML-IDS technique under 2000 epochs and distinct classes.

Table 3. Result analysis of ESOML-IDS technique under 2000 epochs and distinct classes.

Epoch-2000						
Class Labels	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**	**Precision**	**Recall**	**F1-Score**
Normal	81.54	82.64	83.02	81.92	83.49	82.18
DoS	82.72	83.10	84.78	83.08	81.46	83.82
Backdoor	81.63	82.73	82.11	81.47	83.05	81.69
Exploits	82.86	83.70	83.24	83.10	82.82	82.39
Analysis	83.15	83.01	84.93	82.95	81.65	83.46
Generic 83.80	81.09	83.92	82.29	83.19	83.97	
Fuzzers 82.12	82.85	81.29	81.64	81.53	83.92	
Shellcode	83.42	81.37	83.29	83.35	82.15	82.55
Reconnaissance	83.37	82.17	81.51	82.29	82.95	83.04
Worms	82.64	83.59	82.82	82.74	82.68	83.75
Average	**82.73**	**82.63**	**83.09**	**82.48**	**82.50**	**83.08**

The accuracy outcome analysis of the ESOML-IDS approach on test data is showcased in Figure 9. The results demonstrated that the ESOML-IDS technique achieved improved validation accuracy related to training accuracy. It can be also observed that the accuracy values become saturated with the epoch count of 2000. The loss outcome analysis of the ESOML-IDS technique on test data is displayed in Figure 10. The figure reveals that the ESOML-IDS system resulted in reduced validation loss in terms of the training loss. It is additionally noticed that the loss values become saturated with an epoch count of 2000.

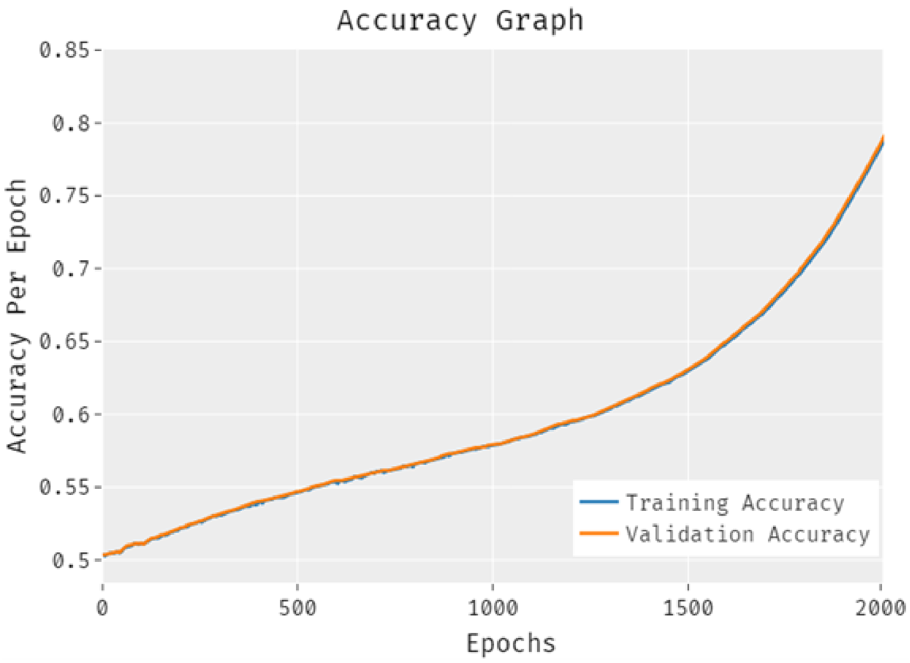

Figure 9. Accuracy analysis of ESOML-IDS technique under 2000 epochs.

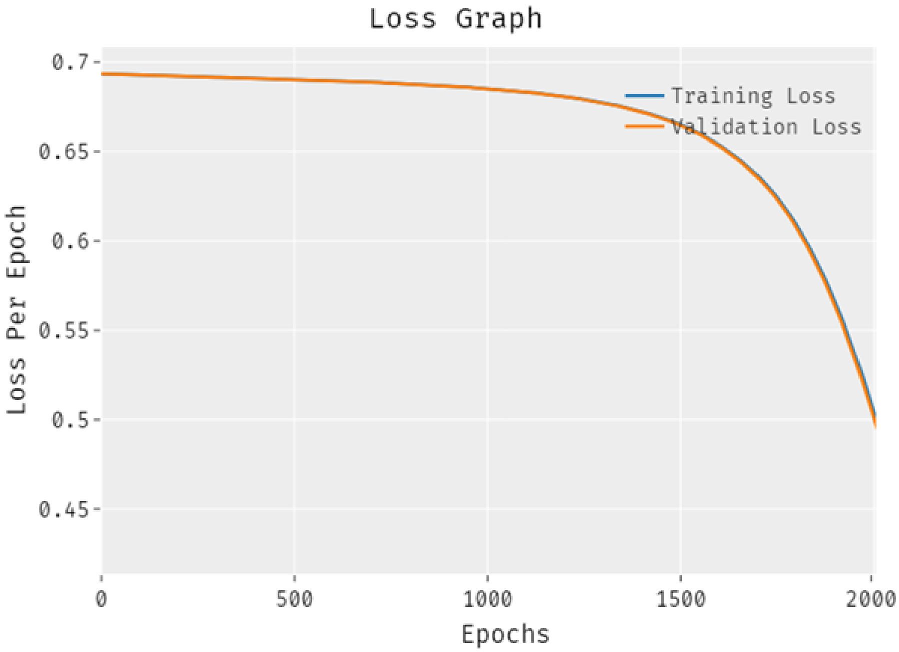

Figure 10. Loss analysis of ESOML-IDS technique under 2000 epochs.

Table 4 and Figure 11 provide a comparative study of the DAE-IDS technique with existing techniques in terms of distinct measures. The results indicated that the SVM technique gained ineffective results, with $accu_y$ of 0.6109, $prec_n$ of 0.4747, $reca_l$ of 0.6200, and $F1_{score}$ of 0.5377. In line with, the LR technique offered somewhat increased outcomes, with $accu_y$ of 0.6553, $prec_n$ of 0.7691, $reca_l$ of 0.6554, and $F1_{score}$ of 0.6662. Then, the DT technique yielded moderate results, with $accu_y$ of 0.6603, $prec_n$ of 0.7982, $reca_l$ of 0.6604, and $F1_{score}$ of 0.5112. Although the ANN and KNN techniques achieved reasonable classification results, the DAE-IDS technique showed enhanced performance, with $accu_y$ of 0.7834, $prec_n$ of 0.8010, $reca_l$ of 0.7786, and $F1_{score}$ of 0.7946.

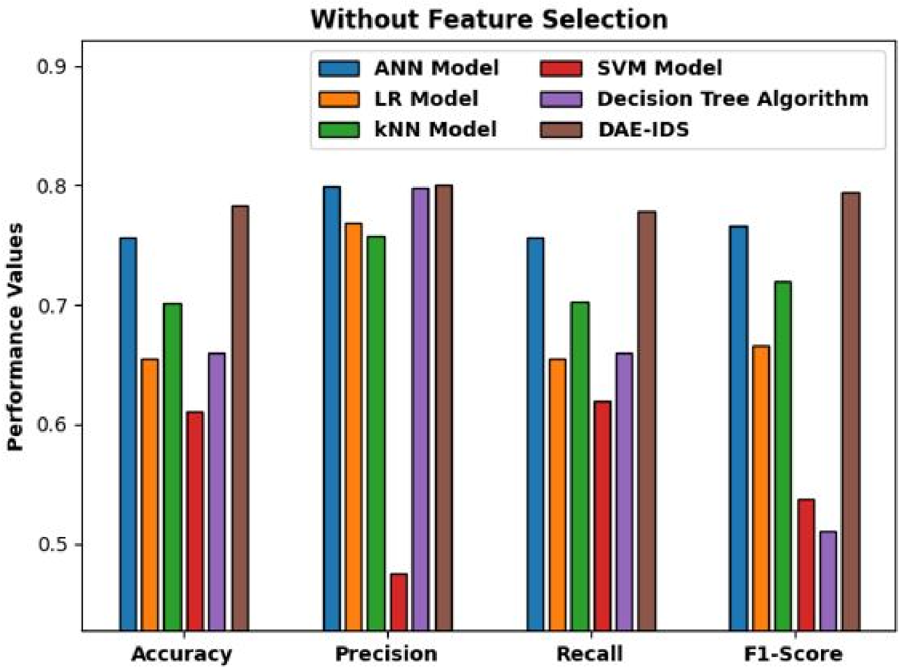

Figure 11. Comparative analysis of DAE-IDS technique without feature selection.

Table 4. Comparative analysis of DAE-IDS technique without feature selection.

Methods	Accuracy	Precision	Recall	F1 Score
ANN Model	0.7562	0.7992	0.7561	0.7658
LR Model	0.6553	0.7691	0.6554	0.6662
kNN Model	0.7009	0.7579	0.7021	0.7203
SVM Model	0.6109	0.4747	0.6200	0.5377
Decision Tree Algorithm	0.6603	0.7982	0.6604	0.5112
DAE-IDS	0.7834	0.8010	0.7786	0.7946

Table 5 and Figure 12 provide a comparative study of the ESOML-IDS model with existing models in terms of distinct measures. The results indicated that the SVM approach yielded ineffectual results, with $accu_y$ of 0.6153, $prec_n$ of 0.5395, $reca_l$ of 0.6152, and $F1_{score}$ of 0.5131. Likewise, the LR system offered slightly increased outcomes, with $accu_y$ of 0.6529, $prec_n$ of 0.7088, $reca_l$ of 0.6529, and $F1_score$ of 0.6569.

Table 5. Comparative analysis of DAE-IDS technique with feature selection.

Methods	Accuracy	Precision	Recall	F1 Score
ANN Model	0.7751	0.7950	0.7753	0.7728
LR Model	0.6529	0.7088	0.6529	0.6596
kNN Model	0.7230	0.7724	0.7230	0.7381
SVM Model	0.6153	0.5395	0.6152	0.5131
Decision Tree Algorithm	0.6757	0.7966	0.6756	0.6926
ESOML-IDS	0.8309	0.8248	0.8250	0.8308

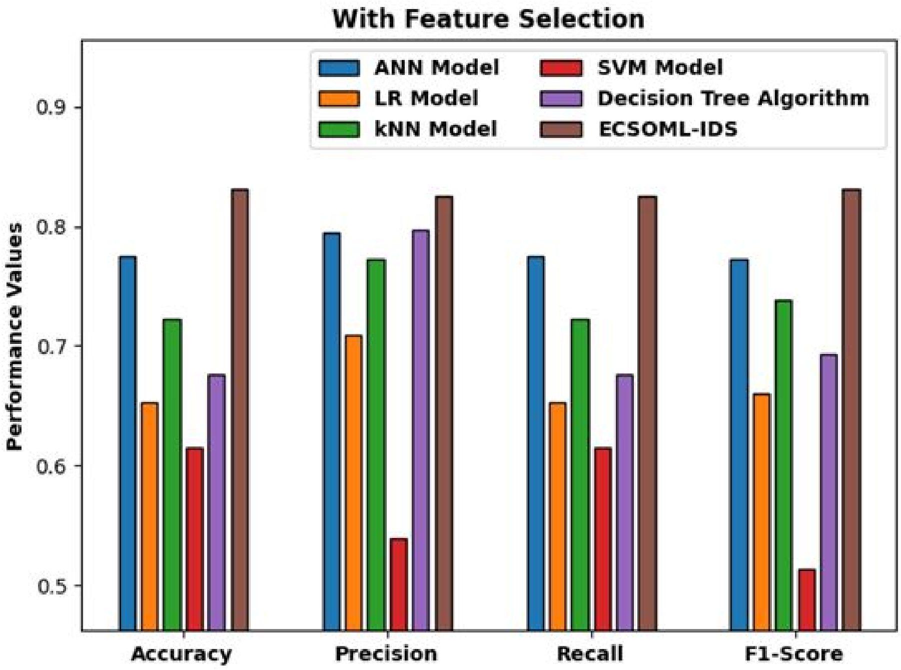

Figure 12. Comparative analysis of DAE-IDS technique with feature selection.

Then, the DT approach yielded moderate results, with $accu_y$ of 0.6757, $prec_n$ of 0.7966, $reca_l$ of 0.6756, and $F1_{score}$ of 0.6926. Afterward, the ANN and KNN models reached reasonable classification results, and the ESOML-IDS method accomplished enhanced performance, with $accu_y$ of 0.8309, $prec_n$ of 0.8248, $reca_l$ of 0.8250, and $F1_{score}$ of 0.8308. After examining the above-mentioned tables and figures, it is apparent that the presented model achieved superior intrusion detection outcomes over the other techniques.

5. Conclusions

For intrusion detection and classification in FC and EC environments, a new ESOML-IDS technique has been developed in this manuscript, aiming to identify the occurrence of intrusions. The ESOML-IDS technique consists of a series of sub-processes including pre-processing, ESO-based feature subset selection, a DAE classifier, and CLPSO-based parameter optimization. For improving the detection efficiency in the aforementioned environments, the ESO algorithm for feature subset selection and DAE for parameter optimization have been utilized. Additionally, to demonstrate the enhanced outcomes of the ESOML-IDS model, a wide variety of empirical experiments with exhaustive simulations were carried out. The experimental results reported the enhanced outcomes of the ESOML-IDS model over the recent approaches, showing the superiority of the proposed technique in terms of accuracy, precision, recall, and F1 score. We believe that the proposed technique can be used to extract manifold benefits with a minimal loss in accuracy for detecting intrusions in FC and EC environments.

Author Contributions: All the authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by O.A.A., J.A.A., and M.A. The first draft of the manuscript was written by A.A. (Adnan Alrabea), A.A. (Albara Awajan), and I.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research and Innovation at Al-Balqa Applied University, Al-Salt, Jordan. Grant Number: DSR-2021#380.

Data Availability Statement: The datasets analyzed during this study are available from the corresponding author upon reasonable request.

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

References

1. Khater, B.S.; Wahab, A.W.A.; Idris, M.Y.I.; Hussain, M.A.; Ibrahim, A.A.; Amin, M.A.; Shehadeh, H.A. Classifier performance evaluation for lightweight ids using fog computing in iot security. *Electronics* **2021**, *10*, 1633. [CrossRef]
2. Onah, J.O.; Abdulhamid, S.M.; Abdullahi, M.; Hassan, I.H.; Al-Ghusham, A. Genetic algorithm based feature selection and naïve bayes for anomaly detection in fog computing environment. *Mach. Learn. Appl.* **2021**, *6*, 100156. [CrossRef]
3. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R. A distributed framework for detecting ddos attacks in smart contract-based blockchain-iot systems by leveraging fog computing. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4112. [CrossRef]
4. Alzubi, O.A.; Qiqieh, I.; Alzubi, J.A. Fusion of deep learning based cyberattack detection and classification model for intelligent systems. *Clust. Comput.* **2022**, 1–12. *in press*. [CrossRef]
5. Zwayed, F.A.; Anbar, M.; Sanjalawe, Y.; Manickam, S. Intrusion detection systems in fog computing- a review. In *Advances in Cyber Security*; Abdullah, N., Manickam, S., Anbar, M., Eds.; Springer: Singapore, 2021; pp. 481–504.
6. Aliyu, F.; Sheltami, T.; Shakshuki, E.M. A detection and prevention technique for man in the middle attack in fog computing. *Procedia Comput. Sci.* **2018**, *141*, 24–31. [CrossRef]
7. Alrawais, A.; Alhothaily, A.; Hu, C.; Cheng, X. Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Comput.* **2017**, *21*, 34–42. [CrossRef]
8. Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. *Energies* **2019**, *12*, 1223. [CrossRef]
9. Alzubi, O.A.; Alzubi, J.A.; Al-Zoubi, A.M.; Hassonah, M.A.; Kose, U. An efficient malware detection approach with feature weighting based on harris hawks optimization. *Clust. Comput.* **2022**, *25*, 2369–2387. [CrossRef]
10. Zong, W.; Chow, Y.-W.; Susilo, W. A two-stage classifier approach for network intrusion detection. In *Information Security Practice and Experience*; Su, C., Kikuchi, H., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 329–340.
11. Alzubi, O.A.; Alzubi, J.A.; Shankar, K.; Gupta, D. Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in internet of things. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4360. [CrossRef]
12. Alazab, M.; Khurma, R.A.; Awajan, A.; Camacho, D. A new intrusion detection system based on moth–flame optimizer algorithm. *Expert Syst. Appl.* **2022**, *210*, 118439. [CrossRef]
13. Alzubi, O.A. Quantum readout and gradient deep learning model for secure and sustainable data access in iwsn. *PeerJ Comput. Sci.* **2022**, *8*, e983–e1007. [CrossRef]
14. Alazab, M.; Layton, R.; Broadhurst, R.; Bouhours, B. Malicious spam emails developments and authorship attribution. In Proceedings of the 2013 Fourth Cybercrime and Trustworthy Computing Workshop, Sydney, Australia, 21–22 November 2013; IEEE: Minneapolis, MN, USA, 2013; pp. 58–68.
15. Kumar, V.; Sinha, D.; Das, A.K.; Pandey, S.C.; Goswami, R.T. An integrated rule based intrusion detection system: Analysis on unsw-nb15 data set and the real time online dataset. *Clust. Comput.* **2019**, *23*, 1397–1418. [CrossRef]
16. Alzubi, O.A. A deep learning- based frechet and dirichlet model for intrusion detection in iwsn. *J. Intell. Fuzzy Syst.* **2022**, *42*, 873–883. [CrossRef]
17. Chen, T.M.; Blasco, J.; Alzubi, J.; Alzubi, O. Intrusion detection. *Eng. Technol. Ref.* **2014**, *1*, 1–9. [CrossRef]
18. Lin, F.; Zhou, Y.; An, X.; You, I.; Choo, K.-K.R. Fair resource allocation in an intrusion-detection system for edge computing: Ensuring the security of internet of things devices. *IEEE Consum. Electron. Mag.* **2018**, *7*, 45–50. [CrossRef]
19. Li, Q.; Hou, J.; Meng, S.; Long, H. Glide: A game theory and data-driven mimicking linkage intrusion detection for edge computing networks. *Complex* **2020**, *2020*, 7136160:1–7136160:18. [CrossRef]
20. Wang, Y.; Meng, W.; Li, W.; Liu, Z.; Liu, Y.; Xue, H. Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5101. [CrossRef]
21. Khater, B.S.; Wahab, A.W.B.A.; Idris, M.Y.I.B.; Hussain, M.A.; Ibrahim, A.A. A lightweight perceptron-based intrusion detection system for fog computing. *Appl. Sci.* **2019**, *9*, 178. [CrossRef]
22. An, X.; Su, J.; Lü, X.; Lin, F. Hypergraph clustering model-based association analysis of ddos attacks in fog computing intrusion detection system. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 249–259. [CrossRef]
23. Mourad, A.; Tout, H.; Wahab, O.A.; Otrok, H.; Dbouk, T. Ad hoc vehicular fog enabling cooperative low-latency intrusion detection. *IEEE Internet Things J.* **2021**, *8*, 829–843. [CrossRef]
24. Abdel-Basset, M.; Chang, V.; Hawash, H.; Chakrabortty, R.K.; Ryan, M. Deep-ifs: Intrusion detection approach for industrial internet of things traffic in fog environment. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7704–7715. [CrossRef]
25. Pacheco, J.; Benitez, V.H.; Félix-Herrán, L.C.; Satam, P. Artificial neural networks-based intrusion detection system for internet of things fog nodes. *IEEE Access* **2020**, *8*, 73907–73918. [CrossRef]
26. Duan, S.; Luo, H.; Liu, H. An elastic collision seeker optimization algorithm for optimization constrained engineering problems. *Math. Probl. Eng.* **2022**, *2022*, 1344667. [CrossRef]
27. Liang, P.; Shi, W.; Zhang, X. Remote sensing image classification based on stacked denoising autoencoder. *Remote. Sens.* **2017**, *10*, 16. [CrossRef]
28. Ji, Y.; Zhao, X.; Hao, J. A novel uav path planning algorithm based on double-dynamic biogeography-based learning particle swarm optimization. *Mob. Inf. Syst.* **2022**, *2022*, 8519708. [CrossRef]