I. J. Computer Network and Information Security, **2024, 2, 79-91**
Published Online on April 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.02.07

Optimized Intrusion Detection System in Fog Computing Environment Using Automatic Termination-based Whale Optimization with ELM

Dipti Prava Sahu*
Biju Patnaik University of Technology / Department of Computer Science and Engineering, Rourkela, Odisha, 769004, India
E-mail: diptiprava29@gmail.com
ORCID iD: https://orcid.org/0009-0004-8777-5243
*Corresponding Author

Biswajit Tripathy
Einstein College of Computer Application and Management / Master of Computer Applications, Khurda, Odisha, 752060, India
E-mail: biswajit69@gmail.com
ORCID iD: https://orcid.org/0000-0001-9707-9170

Leena Samantaray
Ajay Binay Institute of Technology / Department of Electronics and Communication Engineering, Cuttack, Odisha, 753014, India
E-mail: leena_sam@rediffmail.com
ORCID iD: https://orcid.org/0009-0005-8201-096X

Received: 21 June 2023; Revised: 26 August 2023; Accepted: 06 September 2023; Published: 08 April 2024

Abstract: In fog computing, computing resources are deployed at the network edge, which can include routers, switches, gateways, and even end-user devices. Fog computing focuses on running computations and storing data directly on or near the fog devices themselves. The data processing occurs locally on the device, reducing the reliance on network connectivity and allowing for faster response times. However, the conventional intrusion detection system (IDS) failed to provide security during the data transfer between fog nodes to cloud, fog data centres. So, this work implemented the optimized IDS in fog computing environment (OIDS-FCE) using advanced naturally inspired optimization algorithms with extreme learning. Initially, the data preprocessing operation maintains the uniform characteristics in the dataset by normalizing the columns. Then, comprehensive learning particle swarm based effective seeker optimization (CLPS-ESO) algorithm extracts the intrusion specific features by analyzing the internal patterns of all rows, columns. In addition, automatic termination-based whale optimization algorithm (ATWOA) selects the best intrusion features from CLPS-ESO resultant features using correlation analysis. Finally, the hybrid extreme learning machine (HELM) classifies the varies instruction types from ATWOA optimal features. The simulation results show that the proposed OIDS-FCE achieved 98.52% accuracy, 96.38% precision, 95.50% of recall, and 95.90% of F1-score using UNSW-NB dataset, which are higher than other artificial intelligence IDS models.

Index Terms: Fog Computing, Intrusion Detection System, Effective Seeker Optimization Algorithm, Improved Whale Optimization, Hybrid Extreme Learning Machine.

1. Introduction

Fog computing environments are distributed computing architectures that aim to carry calculation and data storage closer to the edge of the network [1,2], where the data is generated or consumed. Communication Infrastructure relies on a robust communication infrastructure to enable data exchange between different layers. This infrastructure includes

wired and wireless networks [3,4], such as Ethernet, Wi-Fi, cellular networks, or specialized IoT protocols [5]. Data generated by fog devices is processed and stored at various levels of the architecture. Simple data processing may occur directly on the fog devices themselves, while more complex analytics and computations are offloaded to fog nodes or the cloud. Data storage is distributed across fog nodes, cloud storage services, or fog devices depending on the specific requirements and constraints [6]. The architecture incorporates data flow and routing mechanisms to efficiently transmit data between layers. Data is routed from fog devices to fog nodes or directly to the cloud, depending on factors such as latency requirements, computational capabilities, and network conditions.

1.1. Problem Statement

The need for security in fog computing environments arises due to several reasons, such as distributed architecture, proximity to the edge, limited resources, network heterogeneity, and privacy concerns. Fog computing involves a distributed network of devices and resources, making it more challenging to manage security compared to a centralized cloud environment [7]. The distributed nature increases the attack surface, as there are more entry points that could potentially be compromised. The proximity of computing resources to the network edge means they are more exposed to physical threats, such as tampering, theft, or unauthorized access [8]. This proximity makes it important to secure the devices themselves to prevent unauthorized manipulation or extraction of sensitive data. Fog devices have limited computational energy resources, memory, and power. So, implementation of IDS based security protocol in these devices is a difficult process. Moreover, limited resources can hinder the deployment of complex security protocols or encryption algorithms [9]. Fog computing environments typically consist of diverse devices, operating systems, communication protocols, and vendors. This heterogeneity introduces compatibility and interoperability challenges, making it difficult to implement consistent security measures across the entire environment. Fog computing involves processing and storing data at or near the source, which raises privacy concerns. Sensitive or personal data is captured and processed at the edge, necessitating strong security measures to safeguard the confidentiality, integrity, and privacy of information. To address these security challenges, several measures are implemented in fog computing environments**.** The IDSs [10] are deployed to monitor the network traffic and detect any suspicious or malicious activity. These systems can help identify and respond to security breaches in real-time.

1.2. Research Goal

The goal of this work is to detect the various intrusions in fog computing. So, the novel contributions are defined to meet the research goal, which are defined as follows:

- Design of OIDS-FCE to address the unique challenges of distributed environments with resource limitations and real-time requirements.
- The work introduces the CLPS-ESO for extracting intrusion-specific features by analyzing internal patterns of dataset rows and columns.
- An ATWOA is employed to select the most relevant intrusion features using natural inspired correlation analysis.
- The HELM is utilized for accurate classification of various intrusion types in fog computing environments.

The rest of the article is organized as follows: section 2 comprises detailed analysis of existing methods such as related work. Section 3 illustrates the operation details of OIDS-FCE with CLPS-ESO feature extraction, ATWOA feature selection and HELM classification. Section 4 gives results of the OIDS-FCE. Section 5 concludes the article with possible future scope.

2. Related Works

Aliyu et al. [11] suggested a lightweight IDS for the fog layer that was anomaly-based, human immune, and based on unusual occurrences. The IDS functionalities are distributed between the fog nodes, which allows for a minimal resource overhead to be achieved. In addition to this, when comparing the deployment of a neural network on the fog node to their approach, they found that their approach resulted in a 10% decrease in the amount of energy that was used by the fog node. Chen et al. [12] offered IDSs using multi-objective evolutionary convolutional neural network (MECNN), which is intended to be operated on the fog nodes. However, this method has lower reliability in terms of security. Labiod et al. [13] presented IDS, which is both distributed and lightweight. Combining variational AutoEncoder with multilayer perceptron is what the suggested IDS. However, this method is suffering with higher synchronization issues. A hybrid model combining stacked autoencoders (SAE) and CNNs was suggested by Telikani et al. [14]. The loss function is developed to optimize the model's parameters, and an evolutionary algorithm is used to compute the costs of various optimization strategies. De Souza et al. [15] developed a strategy consisting of two stages for the detection and identification of intrusions. In the initial part of the process, a traffic analysis using an Extra Tree binary classifier is carried out.

Ramkumar and colleagues [16] created an algorithm called the Rider Sea Lion Optimization (RSLO). An optimization-driven hybrid ensemble classifier was presented by the authors for use in a fog computing environment.

The cloud layer, the end point layer, and the fog layer make up the trio of layers that are used in fog computing to carry out the whole of the processing that is performed. Chang et al. [17] presented a Software-as-a-Service based IDS. Cloud and fog computing have gained significant traction in recent years, offering scalable and flexible solutions for data storage, processing, and application deployment. However, the distributed nature of these environments raises security concerns that must be effectively addressed to ensure confidentiality, integrity, and availability of data and services. Lussi et al. [18] suggested a lightweight specification-based IDS for smart environments. This paper proposes a system architecture that incorporates IoT device monitoring directly within the fog computing layer. By integrating monitoring capabilities at this level, the system aims to address latency challenges and provide effective defence mechanisms against denial-of-service (DoS) attacks targeting the fog and layers above it.

Zhao et al. [19] suggested a lightweight IDS that was based on CNN model. It begins with a one-dimensional sequence after its two-dimensional structure is first reduced to that. In addition, the design requirements of the lightweight computer vision model ShuffleNetV2 was used in the process of improving CNN to make the latter model more lightweight. An anomaly-based IDS was presented by Alzahrani et al. [20] as a means of mitigating the effects of an assault. By designing and putting into action a distributed DoS based IDS system that makes use of a statistical technique that combines three distinct algorithms, an effective anomaly mitigation system has been devised and implemented for the IoT network. In [21] authors proposed a deep auto encoder-based IDS (DAE-IDS) for fog computing environments using a combination of autoencoders and isolation forest. Autoencoders are used for unsupervised feature extraction and dimensionality reduction, while isolation forest is employed for anomaly detection. In [22] authors proposed propose a support vector machine (SVM) based IDS (SVM-IDS) that leverages cloud-fog collaboration. The system employs SVM classifiers at both the fog and cloud layers, utilizing their complementary capabilities. The fog layer performs initial intrusion detection, and the cloud layer carries out the final decision-making process.

In [23] authors presented decision tree classifier-based IDS (DTC-IDS) for fog computing environments. This IDS is designed to handle big data in fog environments, leveraging decision trees for classification and rule generation. In [24] authors developed k-nearest neighborhood-based IDS (KNN-IDS) in fog-based IoT environments. The approach combines signature-based and anomaly-based intrusion detection techniques to improve the overall accuracy and efficiency of intrusion detection. In [25] authors presented an ESO machine learning-based IDS (ESOML-IDS) for fog and edge computing environments. They performed the simulations using ESO based feature extraction with multiple machine learning algorithms. They developed ESO-SVM-IDS, ESO-KNN-IDS, ESO-DTC-IDS combinations, which resulted in poor performance as compared to ESOML-IDS. However, the ESOML-IDS method resulted in reduced classification accuracy due to DAE classifier, and absence of feature selection strategies.

3. Proposed Method

In fog computing, computing resources are deployed at the network edge, which can include routers, switches, gateways, and even end-user devices. This allows for data processing, analysis, and storage to take place closer to where the data is generated. By distributing computation and storage capabilities, fog computing reduces the need to transmit large amounts of data to remote cloud data centres, resulting in lower latency and improved real-time responsiveness. Edge computing focuses on running computations and storing data directly on or near the fog devices themselves. This means that data processing occurs locally on the device, reducing the reliance on network connectivity and allowing for faster response times. Edge computing is especially useful for applications that require real-time analytics, low-latency interactions, or offline capabilities. So, to provide security, the OIDS-FCE provides the optimal security in fog computing environment. Figure 1 shows the proposed OIDS-FCE block diagram. Initially, the data preprocessing incorporates multiple operations, specifically normalization of columns, to maintain uniform characteristics in the dataset. This preprocessing step ensures that the data is in a standardized format, enhancing the effectiveness of subsequent analysis and classification stages. Then, the CLPS-ESO as an optimization algorithm for extracting intrusion-specific features. The CLPS-ESO analyses internal patterns of all rows and columns in the dataset to identify relevant features, improving the accuracy and effectiveness of the IDS. In addition, the ATWOA feature selection operation performed from CLPS-ESO extracted features. Here, ATWOA leverages correlation analysis to identify the most relevant intrusion features from the set of features obtained through CLPS-ESO. This helps enhance the efficiency and performance of the IDS by focusing on the most informative features. Finally, the HWLM is implemented for classifying different intrusion types from ATWOA features. Here, HELM combines the strengths of extreme learning to achieve accurate and efficient classification of various intrusion types in fog computing environments.

3.1. CLPS-ESO Feature Extraction

The CLPS-ESO approach, based on the mathematical model from ESOML-IDS, utilizes the Laplacian UNSW-NB dataset. This dataset has a size of NS×NF, where NS represents the number of samples and NF represents the number of features. The primary objective of the CLPS-ESO problem is to select a subset of features, denoted as S, from the total number of features (NF), with the condition that the size of S is smaller than NF. So, the CLPS-ESO approach leverages the Laplacian UNSW-NB dataset, with its $NS \times NF$ dimensions, to perform feature selection by choosing a smaller subset of features (S) from the larger set of NF features. Figure 2 shows the CLPS-ESO feature extraction

flowchart. Furthermore, Table 1 shows the CLPS-ESO feature extraction algorithm. The fitness function of CLPS-ESO is derived as follows:

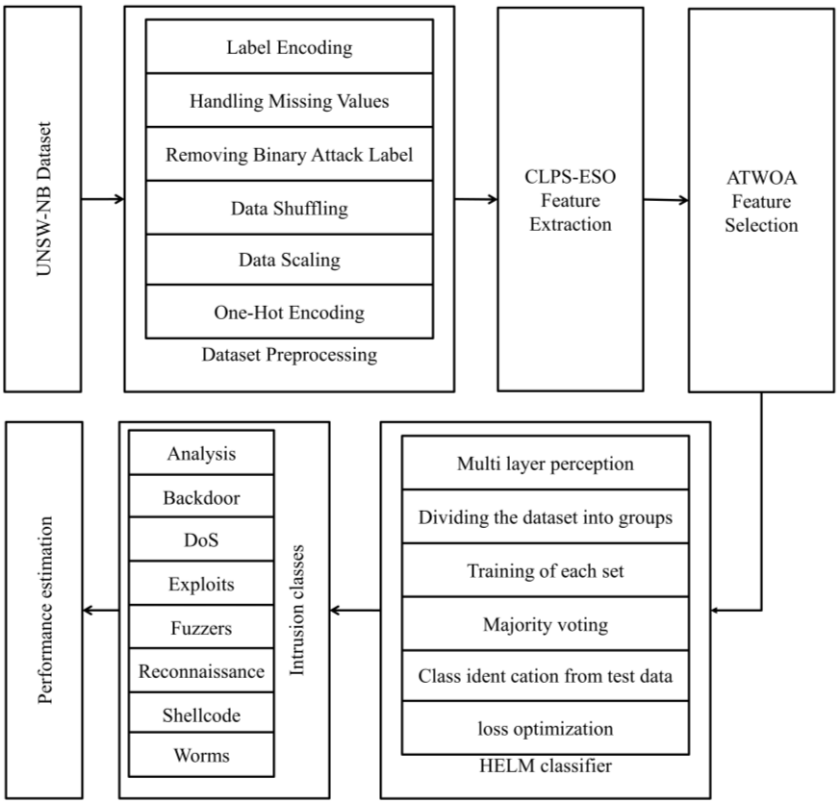


Fig.1. Proposed OIDS-FCE block diagram

$$Fit = \lambda \times \gamma_s + (1 - \lambda) \times \left(\frac{|S|}{NF}\right) \quad (1)$$

In the CLPS-ESO approach, γ_s represents the classifier error when using the selected feature subset S, while $|S|$ denotes the number of features chosen from the total count of NF. The parameter λ is employed to strike a balance between the ratio of $\left(\frac{|S|}{NF}\right)$ and the classifier error γ_s. The incorporation of comprehensive learning in CLPSO allows particles to exchange information, enhancing the cooperative behavior of the swarm. This mechanism enables the particles to learn from each other's experiences, improving the overall exploration and exploitation abilities of the algorithm. The CLPS-ESO strikes a balance between exploration (searching for new solutions) and exploitation (exploiting the current best solutions). By dynamically adjusting the individuals' step size and velocity, ESO improves the algorithm's ability to explore the search space effectively and exploit promising solutions efficiently. This capability helps overcome the problem of getting stuck in local optima, which is a common limitation of some other optimization methods. Further, the combination of CLPS-ESO speeds up the convergence process, allowing the algorithm to find optimal or near-optimal solutions more quickly. ESO's adaptive adjustment of the step size and velocity helps fine-tune the search process, facilitating faster convergence toward the best solutions. The CLPS-ESO exhibits robustness in handling complex optimization problems with high-dimensional search spaces, non-linear relationships, and noisy or incomplete data. Its adaptability and flexibility make it suitable for a wide range of problem domains, including engineering, data science, and machine learning.

3.2. ATWOA Feature Selection

ATWOA is a nature-inspired algorithm that builds upon the properties of the existing WOA algorithm. However, the original WOA algorithm is known to encounter a challenge with local optima, as it lacks the capability to break the optimization loop when further optimization is not possible. To address this limitation, ATWOA incorporates a monitoring mechanism to assess the optimization progress. When no further optimization improvements are achievable, ATWOA intelligently breaks the loop and finalizes the selected features. By incorporating this monitoring mechanism, ATWOA ensures that it does not continue iterating unnecessarily when the optimization process has reached a plateau. This approach prevents the algorithm from getting stuck in local optima, which can hinder the attainment of optimal or near-optimal solutions.

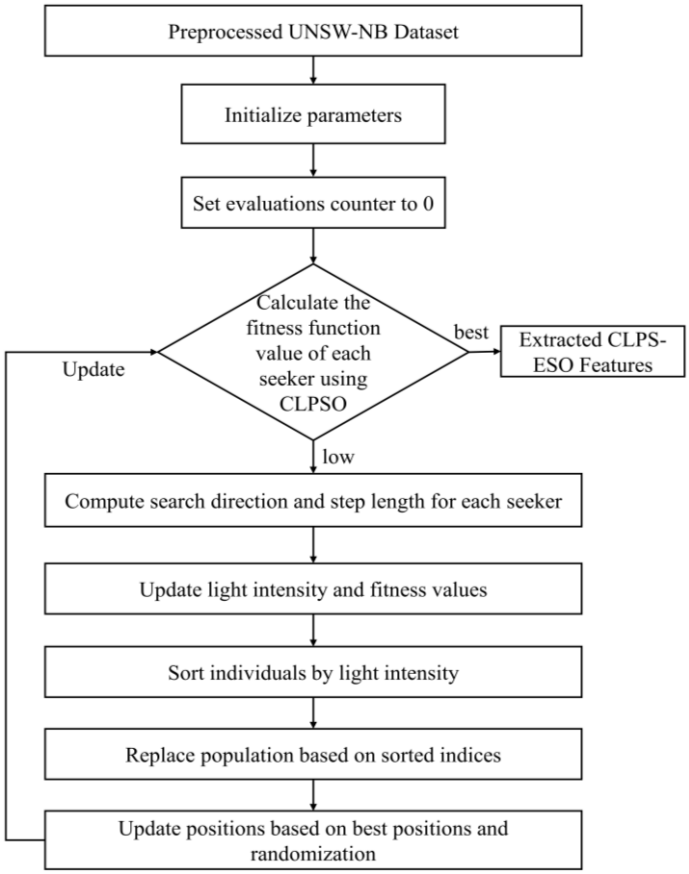


Fig.2. Proposed CLPS-ESO feature extraction flowchart

Table 1. Proposed CLPS-ESO feature extraction algorithm

Input: Preprocessed UNSW-NB Dataset Output: CLPS-ESO extracted features
Step 1: Initialize the CLPS-ESO class with the PSO learning probability objective function. Step 2: Set the algorithm parameters. Step 3: Initialize the population of individuals (ESO agents) with random feature values. Step 4: Initialize fitness, light intensity, and best solution values for everyone. Step 5: Set evaluations counter to 0. Step 6: Repeat until the number of function evaluations reaches nFES. Step 8: Increment the evaluations counter. Step 8.1: Evaluate the fitness of everyone by calling the PSO learning probability objective function. Step 8.2: Update the light intensity values and fitness values of the individuals. Step 8.3: Sort the individuals based on their light intensity values in ascending order. Step 8.4: Replace the old population with the new population based on the sorted indices. Step 8.5: Update the positions of the individuals by considering the best positions and applying randomization. Step 8.6: Check and adjust the feature values of everyone to ensure they are within the defined bounds. Step 9: Return the best individual found.

Figure 3 shows the ATWOA feature selection flowchart. Furthermore, Table 2 shows the ATWOA feature selection algorithm. The ATWOA algorithm is a meta-heuristic optimization technique that imitates the way the brains of humpback whales work. The cortex of the humpback whale's brain is composed of spindle cells, much like the cortex of the human brain. The bubble-net feeding strategy is the only known means by which enormous humpback whales pursue their prey; this technique served as the inspiration for ATWOA. This strategy makes use of a one-of-a-kind pattern that allows for the simultaneous capture of a significant number of fish. When they hear the high-pitched cries of one another, When they are ready, the whales will congregate together and dive down to the school of fish. At the same time, the fish make their way to the surface, where the whales emit the distinctive bubbles in a circle of 9-shaped trail in an upward diminishing spiral around the fishes as a barrier so that the fishes are unable to swim. When the whale leader makes a hunting call, all the whales eventually come to the surface with their mouths gaping wide because of the helix-shaped action they perform. Finally, the fitness function (Fit_{ATWOA}) is evaluated as follows:

$$Fit_{ATWOA} = \vec{X}(t+1) * p * CE + (1-p) * \left(\frac{FS_{ATWOA}}{D_{CLSPO-ESO}}\right) \quad (2)$$

Here, $\vec{X}(t+1)$ represents the updated population of ATWOA, p represents the probability of feature selection, CE represents the classification error, FS_{ATWOA} represents total selected features selected by ATWOA, and $D_{CLSPO-ESO}$ represents the dataset of CLPS-ESO features.

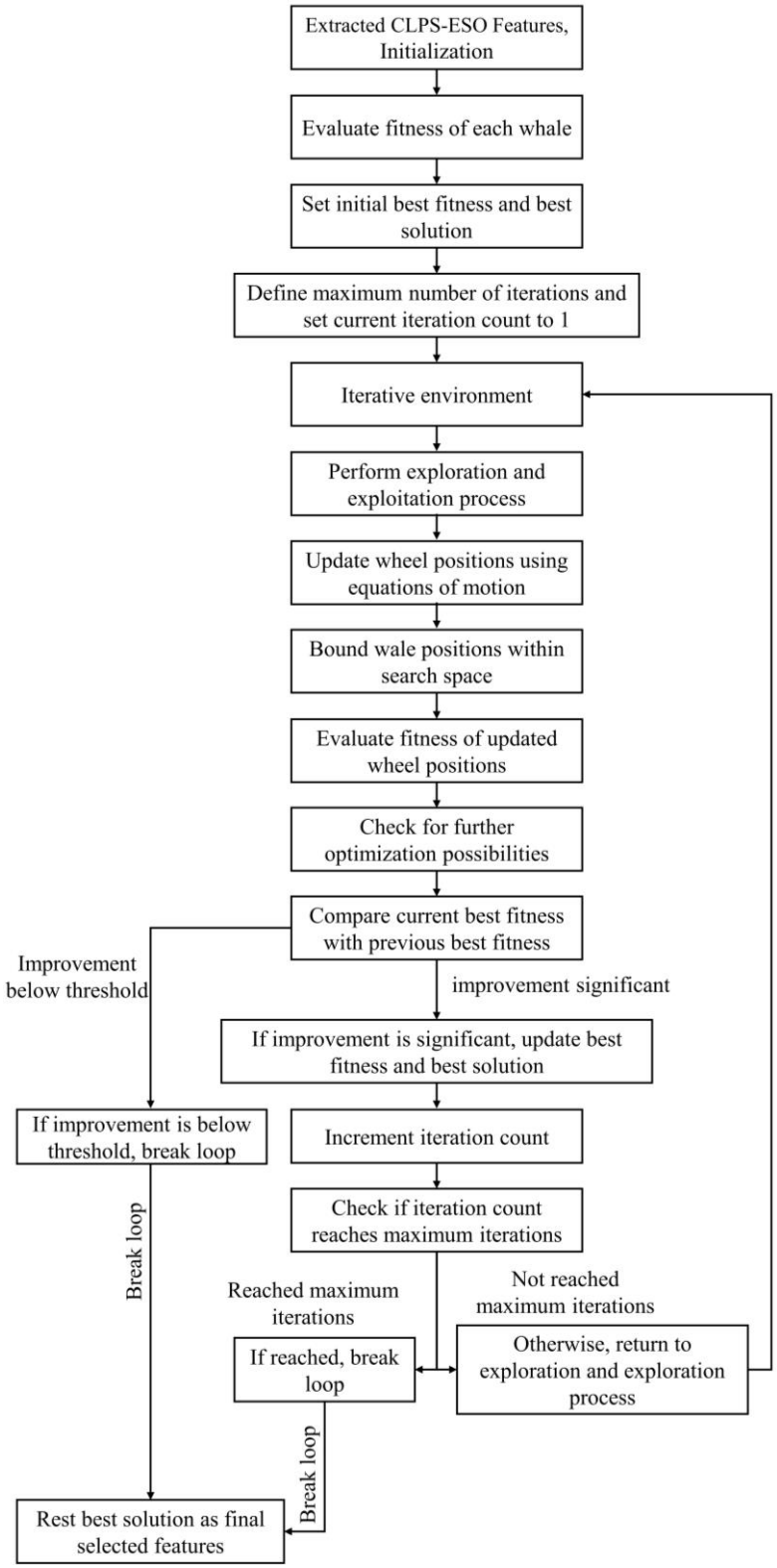


Fig.3. Proposed ATWOA feature selection flowchart

Table 2. Proposed ATWOA feature selection algorithm

Input: CLPS-ESO extracted features Output: ATWOA selected features
Step 1: Initialize the whale population with random solutions. Step 2: Evaluate the fitness of each whale in the population. Step 3: Set the initial best fitness value and the corresponding best solution. Step 4: Define the maximum number of iterations (termination condition) and set the current iteration count to 1. Step 5: Enter the main loop. — Perform the exploration and exploitation process: *Update the positions of the whales using the equations of motion.* *Bound the positions of the whales within the search space if necessary.* *Evaluate the fitness of the updated whale positions.* — Check for further optimization possibilities: — Compare the current best fitness value with the previous best fitness value. *If the improvement is below a predefined threshold or negligible:* *Break the loop and proceed to the next step.* *If the improvement is significant:* *Update the best fitness value and the corresponding best solution.* — Increment the iteration count by 1. — If the iteration count reaches the maximum number of iterations, break the loop. — Otherwise, return to the exploration and exploitation process. Step 6: Return the best solution obtained as the final selected features.

3.3. HELM Classifier

The HELM is a feed-forward neural network (FFNN) widely used for various tasks such as classification, regression, clustering, dimensionality reduction, compression, and pattern learning. One notable advantage of ELMs is their ability to compress data and discover underlying patterns. The operation of HELM is shown in Figure 4, which contains multiple layers of hidden nodes. Importantly, there is no need to modify the characteristics of the hidden nodes, including their biases and weights. The parameters of the hidden nodes can be randomly assigned and remain unchanged indefinitely, or they can be inherited from preceding nodes without any modifications. These models can acquire new information far more quickly than networks that are trained via the process of backpropagation. The learning strategy for backpropagation is the one that is used most often in feed-forward neural networks since it is the most effective learning approach. Here, backpropagation allows the gradients to be computed as they are propagated from the output to the input. Backpropagation is fraught with numerous difficulties. The method of training is quite time intensive in most applications since weights and biases are justified after each iteration of the training process. Because the model ignores the weight magnitude to attain the highest possible level of precision, the result will gradually deteriorate as time goes on. The performance of the learning method for backpropagation is also impacted by the presence of the local minima. The HELM system is a feed-forward network that eliminates the challenge posed by manually adjusting the weights and biases of the system. It is not only focused on reducing the number of training mistakes, but also on achieving the criterion for the least amount of weight feasible, both of which contribute to an improvement in this model's overall level of effectiveness. The issue of becoming stuck in local minima is solved with several straightforward alternatives that sidestep such insignificant concerns.

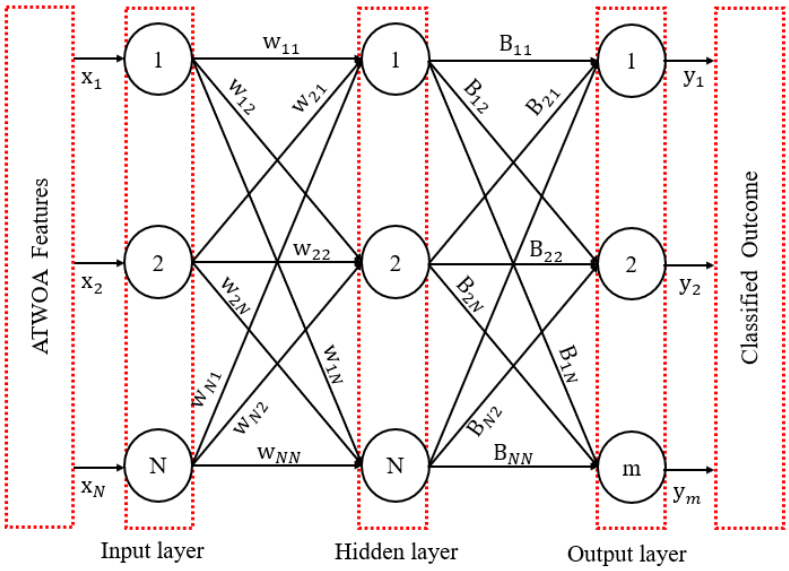


Fig.4. HELM classifier

4. Results

This section gives a detailed analysis of simulation results, which are implemented using UNSW-NB Dataset. Further, the performance of various methods is compared using the same dataset with multiple metrics.

4.1. Dataset

The UNSW-NB computer network security dataset is a comprehensive collection of realistic network activities, encompassing both normal and abnormal behaviors. To obtain this dataset, three virtual servers were utilized in conjunction with an IXIA traffic generator. Two servers were configured to generate normal network traffic, while the third server produced abnormal network traffic. To extract information from the network packets, Argus and Bro-IDS tools were employed. These tools facilitated the extraction of 49 features from the raw network packets. These features encompass both packet-based attributes derived from packet headers and payload, as well as flow-based characteristics obtained by analyzing sequencing, direction, inter-packet length, and inter-arrival times of packets within the network. The dataset's features are divided into several sets: basic features (6 to 18), content features (19 to 26), time features (27 to 35), general-purpose features (36 to 40), and connection features (41 to 47). These features provide information about individual records, connections, and various aspects of network behavior. Moreover, the dataset includes two additional features: attack categories and labels. The attacks are classified into categories such as Analysis, Backdoor, DoS, Exploits, Fuzzers, Reconnaissance, Shellcode, and Worms. It is worth noting that the dataset exhibits significant class imbalance, with normal records representing 87% of the dataset and Worms records accounting for only 0.007%. Overall, the dataset consists of 175,341 records, with 140,272 records allocated for training purposes and 35,069 records for testing.

4.2. Performance Evaluation

Table 3 compares the performance of proposed OIDS-FCE with existing artificial intelligence-based IDS approaches, such as DAE-IDS [21], SVM-IDS [22], DTC-IDS [23], and KNN-IDS [24]. Here, proposed OIDS-FCE shows 20.94% improvement in accuracy, 61.0117% improvement in precision, 102.01% improvement in recall, and 96.0403% improvement in F1-Score compared to DAE-IDS [21]. The proposed OIDS-FCE shows 29.5% improvement in accuracy, 160.9% improvement in precision, 139.55% improvement in recall, and 151.44% improvement in F1-Score compared to SVM-IDS [22]. The proposed OIDS-FCE shown 32.25%, 122.97%, 147.00%, and 114.67% in accuracy, precision, recall, and F1-Score, respectively as compared to DTC-IDS [23]. The proposed OIDS-FCE showed 27.70%, 70.75%, 105.07%, and 97.48% improvement in accuracy, precision, recall, and F1-Score as compared to KNN-IDS [24].

Table 4 compares the performance of proposed OIDS-FCE with existing artificial intelligence approaches, such as ESOML-IDS [25], ESO-SVM-IDS [25], ESO-DTC-IDS [25], and ESO-KNN-IDS [25]. Proposed OIDS-FCE showed a percentage improvement of 19.0473% in Accuracy, 34.2674% in Precision, 84.1348% in Recall, and 73.3663% in F1-Score compared to ESOML-IDS [25]. Proposed OIDS-FCE showed a percentage improvement of 29.2125% in Accuracy, 135.0244% in Precision, 122.9011% in Recall, and 136.3377% in F1-Score compared to ESO-SVM-IDS [25]. Proposed OIDS-FCE showed a percentage improvement of 20.9741% in Accuracy, 50.8663% in Precision, 63.4806% in Recall, and 63.3329% in F1-Score compared to ESO-DTC-IDS [25]. Proposed OIDS-FCE showed a percentage improvement of 23.15% in Accuracy, 94.2927% in Precision, 92.9293% in Recall, and 94.4193% in F1-Score compared to ESO-KNN-IDS [25].

Table 3. Performance comparison of OIDS-FCE with existing AI-IDS approaches

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DAE-IDS [21]	81.4365	59.7817	47.116	48.788
SVM-IDS [22]	76.4	36.97	39.87	38.07
DTC-IDS [23]	74.4	43.39	38.58	44.71
KNN-IDS [24]	77.4	56.38	46.63	48.55
Proposed OIDS-FCE	98.52	96.38	95.50	95.90

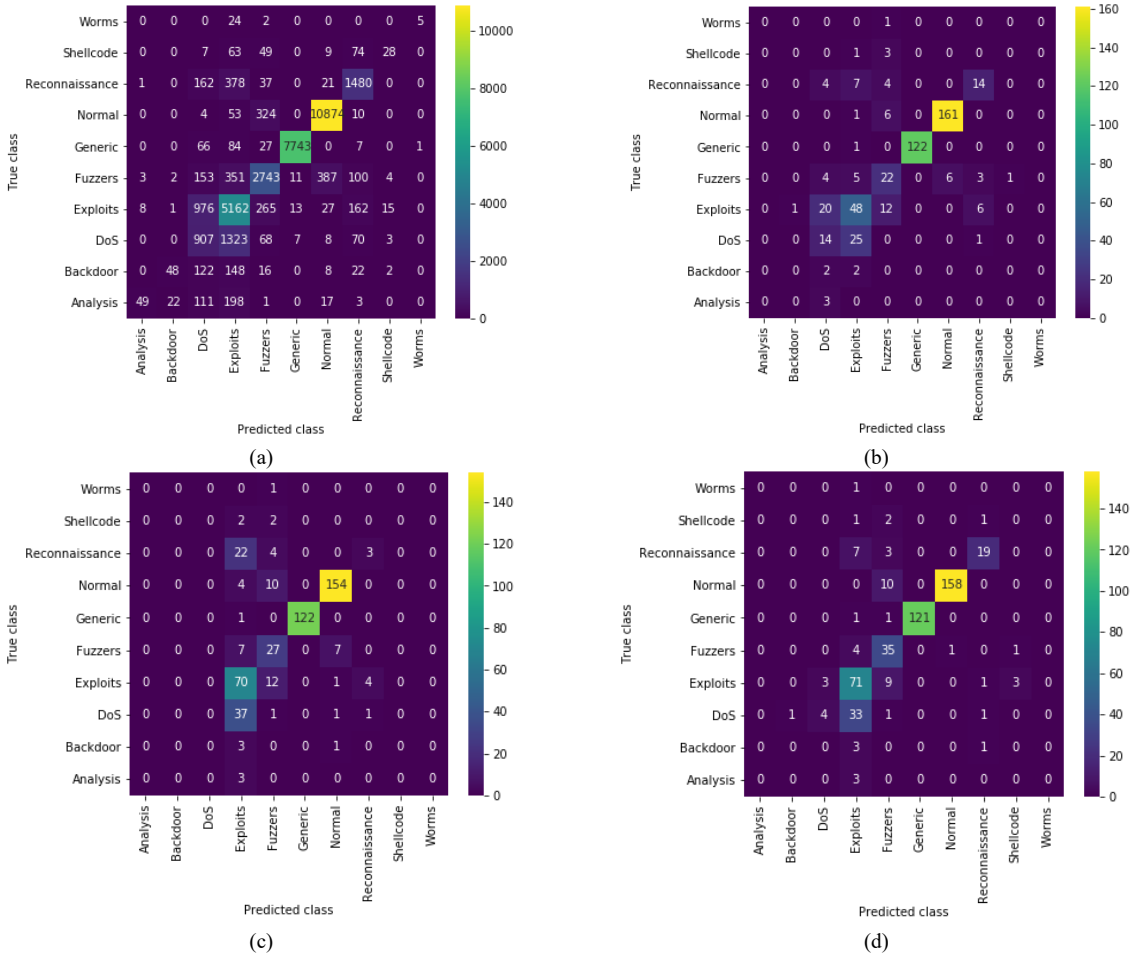
Table 4. Performance comparison of OIDS-FCE with existing optimal artificial intelligence approaches

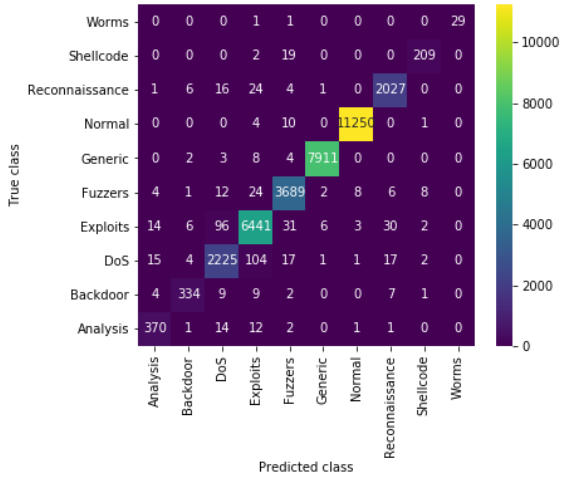
Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ESOML-IDS [25]	82.84	72.03	51.81	55.40
ESO-SVM-IDS [25]	75.8	41.00	42.94	40.60
ESO-DTC-IDS [25]	81.39	63.96	59.02	58.68
ESO-KNN-IDS [25]	80.0	49.67	49.55	49.45
Proposed OIDS-FCE	98.52	96.38	95.50	95.90

Figure 5 shows the confusion matrices of ESOML-IDS [25], ESO-KNN-IDS [25], ESO-SVM-IDS [25], ESO-DTC-IDS [25], and Proposed OIDS-FCE. Here, the proposed OIDS-FCE stands out with higher accuracy, as indicated by its confusion matrix. Specifically, it shows a larger number or percentage of true positives, meaning that the proposed OIDS-FCE effectively identifies and classifies instances of network intrusions correctly. Moreover, the confusion matrix of the proposed OIDS-FCE displays a higher number or percentage of true negatives, indicating its capability to accurately recognize and classify normal network traffic instances. In contrast, the confusion matrices of the other AI-IDS approaches (ESOML-IDS [25], ESO-KNN-IDS [25], ESO-SVM-IDS [25], and ESO-DTC-IDS [25]) demonstrate comparatively lower accuracy. This is reflected by relatively smaller numbers or percentages of true positives and true negatives in their respective confusion matrices. These approaches struggle to accurately identify and classify instances of network intrusions, potentially leading to higher false positives (incorrectly classifying normal traffic as intrusions) and false negatives (failing to recognize actual network intrusions). Therefore, based on the provided confusion matrices, the proposed OIDS-FCE shows superior accuracy compared to the other AI-IDS approaches, indicating its potential as a more effective solution for intrusion detection in terms of correctly classifying both normal and malicious network traffic instances.

Figure 6 shows the region operating characteristic (ROC) curves of various optimal AI-IDS approaches, such as ESOML-IDS [25], ESO-KNN-IDS [25], ESO-SVM- IDS, ESO-DTC-IDS [25], proposed OIDS-FCE. Although ESOML-IDS [25] achieves a decent true positive rate (TPR), it also exhibits a relatively high false positive rate (FPR), leading to a higher number of false positives and potentially more misclassified instances. The ESO-KNN-IDS [25] demonstrates a moderate TPR but suffers from a high FPR, indicating a significant number of false positives, which can result in a less reliable detection system. The ESO-SVM-IDS [25] shows a good balance between TPR and FPR, but it still has a slightly higher FPR compared to the proposed OIDS-FCE, suggesting a relatively higher number of false positives. The ESO-DTC-IDS [25] exhibits a low TPR and a high FPR, indicating a considerable number of false negatives and false positives, respectively, making it less effective in detecting intrusions accurately. The proposed OIDS-FCE approach achieves higher accuracy by striking a better balance between TPR and FPR. It demonstrates a high TPR, effectively capturing true positives, while maintaining a low FPR, resulting in a reduced number of false positives and improving the overall detection performance.

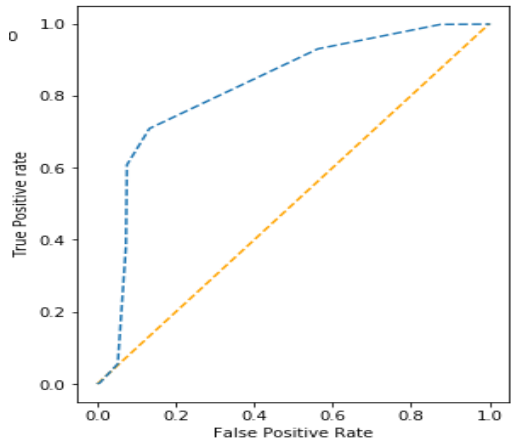
Figure 7 demonstrates the predicted results obtained by the proposed OIDS-FCE on a sample test data from the UNSW-NB dataset. The proposed method is designed to classify different types of attacks such as DoS, Reconnaissance, and Exploits. Additionally, OIDS-FCE can identify instances belonging to the "normal" class in the test data.



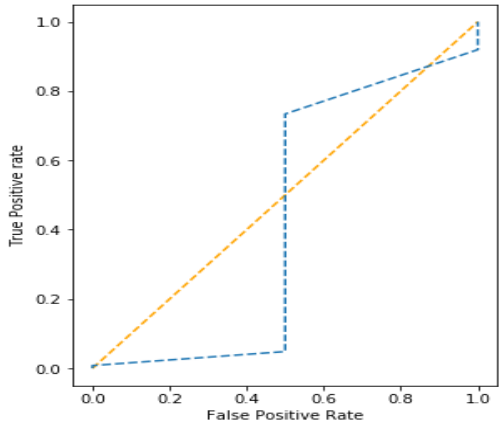


(e)

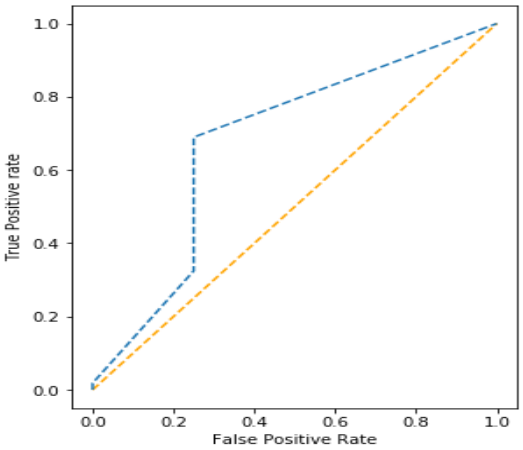
Fig.5. Confusion matrices of various optimal AI-IDS approaches. (a) ESOML-IDS [25]. (b) ESO-KNN-IDS [25]. (c) ESO-SVM- IDS. (d) ESO-DTC-IDS [25]. (e) Proposed OIDS-FCE



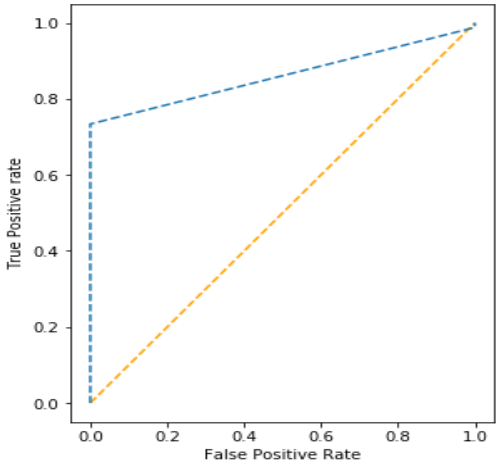
(a)



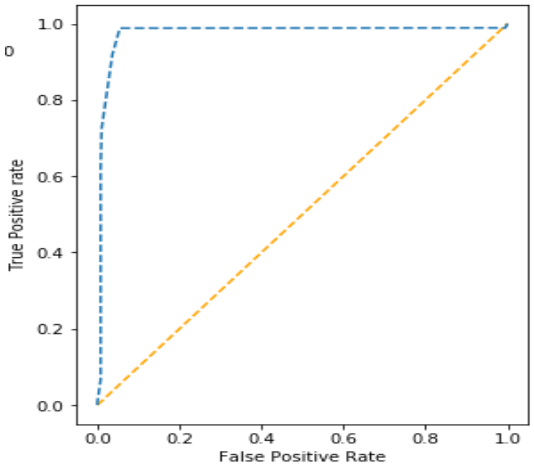
(b)



(c)



(d)



(e)

Fig.6. ROC curves of various optimal AI-IDS approaches. (a) ESOML-IDS [25]. (b) ESO-KNN-IDS [25]. (c) ESO-SVM- IDS. (d) ESO-DTC-IDS [25]. (f) Proposed OIDS-FCE

```
Test Data : [4.04200000e+04 1.65644100e+00 3.00000000e+00 0.00000000e+00
 1.00000000e+00 2.36000000e+02 4.38000000e+02 1.35580000e+04
 5.48216000e+05 4.06292775e+02 3.10000000e+01 2.90000000e+01
 6.52048594e+04 2.64163950e+06 2.10000000e+01 1.97000000e+02
 7.10776800e+00 3.78628400e+00 4.20869209e+02 3.03349756e+02
 2.55000000e+02 8.27280581e+08 8.47673649e+08 2.55000000e+02
 5.81700000e-03 5.61800000e-03 1.99000000e-04 5.70000000e+01
 1.25200000e+03 0.00000000e+00 0.00000000e+00 9.00000000e+00
 0.00000000e+00 4.00000000e+00 2.00000000e+00 1.00000000e+00
 2.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.00000000e+00 3.00000000e+00 0.00000000e+00] Predicted As ====>Exploits
```

Fig.7. Predicted results on sample test data using proposed OIDS-FCE

5. Conclusions

This work has successfully developed OIDS-FCE by leveraging naturally inspired optimization algorithms and extreme learning techniques. Initially, meticulous data preprocessing was performed to ensuring dataset uniformity through normalization. The innovative CLPS-ESO algorithm was then applied to extract intrusion-specific patterns from the data, effectively analysing both rows and columns. Building upon the insights gained from CLPS-ESO, the ATWOA algorithm played a pivotal role in further refining the feature selection process. Through correlation analysis, ATWOA identified and selected the most pertinent intrusion features, enhancing the system's ability to accurately distinguish between normal and intrusive behaviours. The effectiveness of the proposed OIDS-FCE was validated through its integration with the HELM classifier, enabling the system to effectively categorize various intrusion types. The performance evaluation demonstrated remarkable enhancements over existing methodologies, with a substantial percentage improvement of 19.0473% in Accuracy, 34.2674% in Precision, 84.1348% in Recall, and 73.3663% in F1-Score. While this study has achieved significant milestones, there are promising avenues for future research. One such direction involves the exploration and integration of advanced feature selection techniques, which could further elevate the identification of intrusion-specific attributes. Moreover, the potential integration of deep learning-based IDS could usher in higher accuracy levels and extended capabilities.

References

[1] Yi, L., Yin, M., and Darbandi, M., “A deep and systematic review of the intrusion detection systems in the fog environment,” Transactions on Emerging Telecommunications Technologies, vol. 34(1), September 2022.

[2] Hazra, A., Rana, P., Adhikari, M., and Amgoth, T, “Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges,” Computer Science Review, vol. 48, May 2023.

[3] Das, R., and Inuwa, M. M, “A review on fog computing: issues, characteristics, challenges, and potential applications,” Telematics and Informatics Reports, vol. 10, June 2023.

[4] Aqib, M., Kumar, D., and Tripathi, S., “Machine Learning for Fog Computing: Review, Opportunities and a Fog Application Classifier and Scheduler,” Wireless Personal Communications, vol. 129(2), pp. 853-880, December 2022.

[5] Chakraborty, A., Kumar, M., and Chaurasia, N, “Secure framework for IoT applications using Deep Learning in fog Computing,” Journal of Information Security and Applications, vol. 77, Sept. 2023.

[6] Azizpour, S., and Majma, M., “Nada: new architecture for detecting dos and ddos attacks in fog computing,” Journal of Computer Virology and Hacking Techniques, vol. 19(1), pp. 51-64, March 2023.

[7] Yao, W., Shi, H., and Zhao, H., “Scalable anomaly-based intrusion detection for secure Internet of Things using generative

adversarial networks in fog environment," Journal of Network and Computer Applications, vol. 214, May 2023.
[8] Hussein, W. N., Hussain, H. N., Hussain, H. N., and Mallah, A. Q., "A deployment model for IoT devices based on fog computing for data management and analysis," Wireless Personal Communications, pp. 1-13, Feb 2023.
[9] Syed, N. F., Ge, M., and Baig, Z., "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks," Computer Networks, vol. 225, April 2023.
[10] Bukhari, O., Agarwal, P., Koundal, D., and Zafar, S., "Anomaly detection using ensemble techniques for boosting the security of intrusion detection system," Procedia Computer Science, vol. 218, pp. 1003-1013, Jan 2023.
[11] Aliyu, F., Sheltami, T., Deriche, M., and Nasser, N., "Human immune-based intrusion detection and prevention system for fog computing," Journal of Network and Systems Management, vol. 30, pp. 1-27, October 2021.
[12] Chen, Y., Lin, Q., Wei, W., Ji, J., Wong, K. C., and Coello, C. A. C., "Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in Fog computing," Knowledge-Based Systems, vol. 244, May 2022.
[13] Labiod, Y., Korba, A. A., and Ghoualmi, N., "Fog computing-based intrusion detection architecture to protect IoT networks," Wireless Personal Communications, vol. 125(1), pp. 231-259, Feb 2022.
[14] Telikani, A., Adhikari, M., Pourzandi, M., and Zohrevand, A., "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," IEEE Internet of Things Journal, vol. 9(22), pp. 23260-23271, Nov 2022.
[15] De Souza, C. A., Westphall, C. B., and Machado, R. B., "Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments," Computers & Electrical Engineering, vol. 98, March 2022.
[16] Ramkumar, M. P., Ramprakash, T., Subramaniam, S., and Natarajan, P., "Intrusion detection using optimized ensemble classification in fog computing paradigm," Knowledge-Based Systems, vol. 252, September 2022.
[17] Chang, V., Li, J., Li, Y., Yan, X., Zhang, J., and Sun, Q., "A survey on intrusion detection systems for fog and cloud computing". Future Internet, vol. 14(3), March 2022.
[18] Lussi, E. W., Silva, J. L., Sanz, J. L., and Souza, F. B., "A lightweight fog-based internal intrusion detection system for smart environments," International Journal of Intelligent Internet of Things Computing, vol. 1(4), pp. 287-299, Feb 2023.
[19] Zhao, G., Wang, Y., and Wang, J., "Lightweight Intrusion Detection Model of the Internet of Things with Hybrid Cloud-Fog Computing," Security and Communication Networks, vol. 2023, pp. 1-11, Jan 2023.
[20] Alzahrani, R. J., and Alzahrani, A., "A novel multi-algorithm approach to identify network anomalies in the IoT using Fog computing and a model to distinguish between IoT and Non-IoT devices," Journal of Sensor and Actuator Networks, vol. 12(2), February 2023.
[21] Sadaf, K., and Sultana, J., "Intrusion Detection Based on Autoencoder and Isolation Forest in Fog Computing". IEEE Access, vol. 8, pp. 167059-167068, Sept 2020.
[22] Du, R., Li, Y., Liang, X., and Tian, J., "Support vector machine intrusion detection scheme based on cloud-fog collaboration," Mobile Networks and Applications, vol. 27(1), pp. 431-440, January 2022.
[23] Peng, K., Leung, V., Zheng, L., Wang, S., Huang, C., and Lin, T., "Intrusion detection system based on decision tree over big data in fog environment", Wireless Communications and Mobile Computing, vol. 2018, Mar 2018.
[24] De Souza, C. A., Westphall, C. B., Machado, R. B., Sobral, J. B. M., and dos Santos Vieira, G., "Hybrid approach to intrusion detection in fog based IoT environments," Computer Networks, vol. 180, Oct 2020.
[25] Alzubi, O. A., Alzubi, J. A., Alazab, M., Alrabea, A., Awajan, A., and Qiqieh, I., "Optimized machine learning-based intrusion detection system for fog and edge computing environment," Electronics, vol. 11(19), Sept. 2022.

Authors' Profiles

Ms. Dipti Prava Sahu holds B-Tech, M-Tech in Computer Science and Engineering. She has more than 17 years of teaching experience. Currently she is a research scholar in the Department of Computer Science and Engineering, Biu Patnaik University of Technology, Rourkela, Odisha. Her area of interest is Data mining, cloud computing, fog computing, networks, and artificial intelligence.

Dr. Biswajit Tripathy holds a B.E, M-Tech & Ph.D. in Computer Science and Engineering Degrees. Having more than 27 years of experience in Teaching-Learning, Research, Training, Administration, Quality-Improvement, and Industry. Presently he is working as a Professor, ECCAM, Khurda, Odisha. In his crowning glory he has achievements like publishing 4 patents, 2 books, 5 book chapters, and more than 20 international & national research papers. He has received Rs.4,47,500 from BPUT, Odisha for organizing International Conference and Rs.2,59,700 from BPUT, Odisha under CRIS (Collaborative Research and Innovation Scheme) on "Trend analysis of e-commerce website using deep learning". He has attended more than 20 seminars, 15 workshops, 18 QIP/STTP/FDP and 9 webinars. He has delivered several guest lectures across India. He has organized more than 9 Seminar/Workshop/ Training/ FDP Program for the benefit of the society.

Dr. Leena Samantaray holds a B.E, M-Tech, and PhD in Communication Engineering. She is an experienced educationalist with over 18 years of experience in the field of education. Currently she is working as a Professor & Principal, ABIT, Cuttack, Odisha She is a dynamic leader with innovative approach and administrative prowess – determined to produce young Technocrats and Engineering Professionals who would contribute to position India as a technological global hub. She has organized numerous national seminars and student workshops. She has 31 research papers to her credit, which have been published in different National and International Journals, Books & Events.

How to cite this paper: Dipti Prava Sahu, Biswajit Tripathy, Leena Samantaray, "Optimized Intrusion Detection System in Fog Computing Environment Using Automatic Termination-based Whale Optimization with ELM", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.2, pp.79-91, 2024. DOI:10.5815/ijcnis.2024.02.07