ISSN: 2093-5374 / E-ISSN: 2093-5382

Intrusion Detection System Using Chaotic Walrus Optimization-based Convolutional Echo State Networks for IoT-assisted Wireless Sensor Networks

Dr. Salem Bahmaid[1*], and Dr. Sulaiman Abdo Mahyoub Ghaleb[2]

[1*]Faculty of Computer Studies, Arab Open University, Saudi Arabia. s.bahmaid@arabou.edu.sa, https://orcid.org/0009-0009-8649-4470

[2]Faculty of Computer Studies, Arab Open University, Saudi Arabia. s.ghaleb@arabou.edu.sa, https://orcid.org/0000-0003-4737-6876

Received: April 22, 2024; Revised: June 30, 2024; Accepted: August 08, 2024; Published: September 30, 2024

Abstract

Wireless Sensor Networks (WSN) based Internet of Things (IoT) networks have achieved greater research interest due to their multi-purpose data collection and transmission over different geographical locations. However, the fabrication of different cyber-attacks in these networks has been a severe concern for applications such as remote healthcare, military communications, etc. These attacks question the integrity and security of WSN-IOT networks for such applications, and the traditional Intrusion Detection Systems (IDS) have shown increased false alarm rates because of their substantial processing overhead, resource constraints, and low detection rates. This paper presents an intelligent IDS model using Chaotic Walrus Optimization-based Convolutional Echo State Networks (CWO-CESN) to solve existing problems. It increases the detection accuracy of different attacks. In this CWO-CESN-based IDS model, the data are gathered from the sensor/IoT nodes and are pre-processed. Then, the proposed CWO-CESN learns the features from these data and classifies them into attacks and standard data classes. This proposed CWO-CESN is a hybrid classifier model that integrates Convolutional Neural Networks (CNN) and Echo State Networks (ESN) as a single model and employs Chaotic map-based population initialized Walrus Optimizer for optimizing the hyperparameters. Validated on benchmark datasets, the proposed CWO-CESN-based IDS model attained accuracies of 99%, 99.5%, and 99.8% for the detection of different attacks for NSL-KDD, WSN-DS and IoT-23 datasets, respectively, and ensured secured and reliable application in significant fields.

Keywords: Wireless Sensor Networks, Internet of Things, Intrusion Detection System, Convolutional Echo State Networks, Chaotic Walrus Optimization, Flooding Attacks.

1 Introduction

WSN is popularly known as a specialized subset of wireless networks designed to facilitate collecting, preparing, and communicating data from multiple spreader sensor nodes deployed in various domains. IoT allows devices to communicate with each other and users, creating a vast ecosystem of intelligent, connected systems. IoT standards and protocols allow WSNs to communicate with many other IoT devices and systems (Ghaleb & Varadharajan, 2020). IoT enables WSNs to scale more efficiently in

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 15, number: 3 (September), pp. 236-252. DOI: *10.58346/JOWUA.2024.I3.016*

*Corresponding author: Faculty of Computer Studies, Arab Open University, Saudi Arabia.

terms of the more significant number of nodes. These networks are characterized by their ability to operate independently, often in resource-constrained environments, to monitor physical or environmental parameters. WSN leverages low-power, multi-hop communication protocols to support scalability, robustness, and energy efficiency. This network is widely utilized to provide real-time monitoring, remote data collection, and scalability (Ba Hmaid & Varadharajan, 2020). Yet, the WSNs are resource-constrained and distributed networks vulnerable to cyber-attacks and intrusions. For the security challenges, the IDS techniques were integrated with WSN-IoT networks to enhance security by monitoring network traffic and sensor data to detect suspicious behaviour and potential attacks in real-time (Ba Hmaid & Vasanthi, 2020). This integration ensures data integrity and privacy, effectively manages threats, and maintains security compliance.

Conventional methods such as signature and rule-based detection methods are utilized in IDS for WSN-IoT (Ram & Chakraborty, 2024). The signature-based techniques, including pattern matching, detect the threats by comparing the attack signatures to the collected database signature to identify the intrusions. The rule-based method applies rules to detect suspicious behaviour based on known attack patterns. Yet, these models have high false positive rates and struggle with zero-day and unknown attacks (Khedr et al., 2020). The drawbacks of the conventional methods are resolved using DL and ML techniques. The DL methods such as RNN, CNN, LSTM, and Autoencoder provide advanced capabilities in anomaly detection by learning complex mappings from large-scale data (Camgözlü & Kutlu, 2023). The ML methods, namely Decision tree (DT) and random forest (RF), have increased the precision in the classification and detection of diverse attack types (Adnan et al., 2021). However, class imbalance issues, over-fitting, high latency, and ineffective hyperparameter optimization frequently hinder the DL and ML models. A hybrid Chaotic Walrus Optimization-based Convolutional Echo State Networks (CWO-CESN) model is developed to overcome these limitations.

The developed CWO-CESN model incorporates several critical steps for effective intrusion detection. Initially, datasets are gathered from reputable sources, including NSL-KDD, WSN-DS, and IoT-23 (Hui et al., 2019). The data pre-processing steps involve KNN imputation to address the missing values and Min-Max normalization for data scaling, with the SMOTE method applied for class imbalance. The CESN model integrates CNN and Echo State Networks (ESN) in a single architecture, establishing their complementary strengths for enhanced intrusion detection. The CNN model excels in identifying patterns and correlations from the pre-processed data that are used for detecting abnormal events, and the ESN model is used for capturing the temporal dependencies of the network traffic with computational efficiency. The CESN model integrates the ability of CNN and the ability of ESN for spatial feature capturing and temporal feature modeling. The CWO algorithm is used to optimize the hyperparameters, such as the weight and bias of the CESN model. The CWO model combines the strength of chaos theory and walrus optimizer to increase the detection accuracy.

2 Related Works

Ramana et al., (2022) proposed a model using a Whale Optimized Gate Recurrent Unit (WOGRU) ID System for WSN-IoT networks (Skarmeta et al., 2015). The data pre-processing steps include data transformation and data labeling. The processed data are fed into GRU, and the WO technique optimizes hyperparameters. The model has experimented on the WSN-DS, and the WOGRU dataset achieved a 99.8% accuracy, precision, specificity, recall, and F1-score. This model does not incorporate defensive measures but has a robust encryption scheme to defend against attacks. Krishnan et al., (2022) proposed a model using anomalous intrusion detection protocol (AIDP) and intrusion prevention protocol (IPP).

The proposed AIDP applied includes three stages. The nodes update their experience values based on the tiny attack and fault detection system (TAFDS). The experience values are exchanged between neighboring nodes, and the nodes update their reputation and trust values using historical data and newly received experience values. The proposed model used an NS-2 simulator and attained 24% E2E delay and 30% PDR. However, updating and exchanging experience values introduce communication overhead and increase the computational load on nodes. Subramani & Selvi, (2023) proposed a system designed by combining an Intelligent Multi-Objective Particle Swarm Optimization feature selection algorithm for developing the Intrusion Detection System (IMOPSO-IDS). The proposed model used the IMOPSO algorithm and MSVM classifier to select optimal features, and MSVM uses the features chosen for classification. The proposed model attained 99.92% precision, 97.47% recall, 97.72% F-measure, and a training time of 25.24 seconds. However, the IMOPSO algorithm is computationally intensive for large datasets and complex feature spaces. Aljebreen et al., (2023) proposed a model using a Binary Chimp Optimization Algorithm with Machine Learning Intrusion Detection (BCOA-MLID). The features are extracted and selected using the BCOA algorithm based on fitness values. The selected features are given to MSVM with a Class-specific-Cost Regulation Extreme Learning Machine (CCR-ELM). Experiments were conducted using the WSN-DS dataset, and the proposed model achieved 99.63% accuracy, 97.91% sensitivity, 99.67% specificity, and 94.52% F-score. This model is not able to detect the Sybil attack and routing attack.

Alruwaili et al., (2023) proposed a model using the Red Kite Optimization Algorithm with an Average Ensemble Model for Intrusion Detection (RKOA-AEID) technique. The features are extracted and selected using RKOA. The selected features are fed into the developed model, consisting of LSTM, BiLSTM, and BiGRU models. Experiments were conducted on the WSN-DS dataset, and the model achieved 98.94 % accuracy, 75.33% sensitivity, 96.45% specificity, and 79.52% F-score. Yet, the model has a trade-off between detecting attacks and correctly identifying non-attacks. Al Sawafi et al., (2023) proposed a model using a Deep Auto Encoder (DAE) and Deep Artificial Neural Network (DANN). The data are applied to the DAE-DANN model. For DANN architecture, the Multilayer Perceptron (MLP) is adopted. Experiments are conducted on the IoTR-DS dataset, and the proposed model achieved 98% accuracy, 92% precision, 92% recall, and 92% F1 score. Although efficient, the proposed model still suffers from detecting rank attacks. Altunay & Albayrak, (2023) proposed an optimized hybrid deep neural network (OHDNN) with enhanced conditional random field (ECRF). The ECRF algorithm developed CRF and CS for feature selection, which was applied to the OHDNN model for classification, combining attention mechanism, CNN classifier, and LSTM. For the NSL-KDD dataset, the model attained 97.17% accuracy, 97.32% precision, 97.02%recall, 95.92% F-measure, and 14.8% FPR. For the UNSW-NB15 dataset, the model attained 98.3% accuracy, 97.5% precision, 96.7% recall, 97.1% F-measure, and 6.1% FPR (Jelena & Srđan, 2023). However, the model's performance and efficiency decrease when scaled to larger and more complex networks. Yao et al., (2023) proposed a lightweight, intelligent Network Intrusion Detection System (NIDS) using a one-class bidirectional Gated Recurrent Unit (GRU) autoencoder called, (Bi-GRU-AE) and ensemble learning. The OC-Bi-GRU-AE and EL are applied for classification, and the EL model balances the data. The proposed model attained 99.34% accuracy, 99.34% precision, 99.34% recall, and 99.34% F1 score on the WSN-DS dataset. The model attained 90.74% accuracy, 90.65% precision, 90.73% recall, and 90.11% F1-score for the UNSW-NB15 dataset. On the KDD99 dataset, the model achieved 99.99% accuracy, 99.99% precision, 99.99% recall, and 9.99% F1 score. However, combining Bi-GRUs with an autoencoder and ensemble learning techniques results in high computational and memory requirements.

Narayanan et al., (2023) proposed a particle swarm optimization-based artificial neural network (PSO-ANN) model. The Monte Carlo method is utilized for feature extraction. The extracted features are applied to the PSO-ANN model, the ANN parameters are optimized using PSO, and the final classification result is obtained from the ANN model. The proposed model attained 90% accuracy and RMSE of 29%. However, the PSO gets trapped in local optima, particularly in high-dimensional spaces, limiting optimization effectiveness. Karthic & Kumar, (2023). proposed a model using a hybrid Convolutional Neural Network (CNN) with LSTM. The processed data are fed into the CNN-LSTM model for feature learning and classification. Experiments are conducted on two datasets, X-IIoTID and UNSW-NB15. For the UNSW-NB15 dataset, PSO-ANN achieved 92.90% accuracy, 92.91% precision, 93.10% recall and 93% F1-score. Similarly, the X-IIoTID dataset achieved 99.80% accuracy, 99.67% precision, 99.5% recall and 99.6% F1-score. In this model, there is a lack of external feature selection techniques, which may increase the computational load. Awotunde et al., (2023) proposed an ensemble model enabled with a feature selection classifier. A chi-square statistical model is used for feature selection. The selected features are fed into the proposed ensemble models, which include XGBoost, bagging classifier, AdaBoost, extra trees, and RF for classification. The proposed model used a combined IoT dataset and attained 99% accuracy, 99.95% precision, 99.79% recall, and 99.75% F1 score for the XGBoost model. However, XGBoost and RF can be computationally intensive, which may limit their scalability.

Vembu & Ramasamy, (2023) proposed an intrusion detection system (IDS) that incorporates the convolutional neural network (CNN) model. The CNN selects the most relevant features. The whale optimization algorithm fine-tunes the CNN parameters to reduce false alarms and improve detection accuracy. For evaluation, CNN used the NSL-KDD dataset and attained 97% precision, 94.09% recall, 95.50% F1-score, and 97.01% accuracy. However, the performance heavily relies on CNN's ability to select relevant features, which may not always generalize well to unseen or varied intrusions. Biswas et al., (2023). proposed a model using graph neural network (GNN) and Lyapunov optimization in wireless sensor networks for intrusion detection. The data was constructed into graph form and applied to the graph neural network (GNN) for the training process. The proposed model used the AWID dataset and attained 98.80% accuracy, 98.36% sensitivity, 99.14% precision, and 98.74% F1 score. However, Lyapunov optimization leads to additional computational overhead, increasing training time and resource requirements. Karthikeyan et al., (2024) proposed a novel and automated firefly algorithm with machine learning (FA-ML) technique. Features are selected using the Firefly algorithm and fed to the SVM for intrusion detection. The proposed model used the NSL-KDD dataset and attained 96.42% accuracy, 95.35% sensitivity, 98.36% specificity, 95.42%F1-score, and 96.48% AUC score. However, the GWO improves parameter tuning and introduces additional computational complexity and time. Bukhari et al., (2024) proposed a novel federated learning model with a stacked convolutional neural network and bidirectional long short-term memory (FL-SCNN-Bi-LSTM). The principle component analysis (PCA) is utilized for feature selection, and selected features are fed into the FL-SCNN-Bi-LSTM model to classify attack nodes. The model achieved an accuracy of 0.997, F1-score of 0.996, precision of 0.998, and recall of 0.996 for the WSN-DS dataset, and accuracy of 0.9993, F1-score of 0.9993, precision of 0.9993, and recall of 0.9992 for CIC-IDS-2017 dataset. The model can handle more complex networks but still struggles with increased data loads and computational costs.

Talukder et al., (2024) proposed an innovative intrusion detection approach that integrates machine learning (ML) techniques with the synthetic minority oversampling technique Tomek link (SMOTE-TomekLink) algorithm. SMOTE-TomekLink is applied for data balancing. Then, the processed data was used to propose ML techniques, including RF, DT, LightGBM (LGB), KNN, MLP,

and XGB, for final classification. The proposed model used the WSN-DS dataset for evaluation. The RF model attained 99.92% accuracy, 99.92% precision, 99.92% recall, 99.92% F1-score, 11%, MAE of MSE of 17%, and RMSE of 4.15%. The XGB model attained 99.84% accuracy, 99.84% precision, 99.84% recall, 99.84% F1-score, MAE of 23%, MSE of 38%, and RMSE of 6.17%. However, the model has a high computational cost. Yaras & Dener (2024) proposed a one-dimensional Convolutional Neural Network and Long Short-Term Memory (1D-CNN-LSTM) model. The features are extracted and selected using the Pearson correlation coefficient method. The selected features are fed into the 1D-CNN-LSTM model for feature learning and classification. For the CICIoT2023 dataset, the model attained 99.96% accuracy, recall, precision, and F1 score. The TON_IoT dataset achieved 98.75% accuracy, precision, recall, and F1 score. The model's scalability for larger datasets is not addressed, and the model lacks in a dynamic environment. Ravindran, (2024) proposed a novel fuzzy logic-based intrusion detection system with a hidden Markov model (FIDS-HMM). The proposed FIDS-HMM model applied, the HMM estimates each HS's probability based on observed energy consumption and identifies high and low energy levels. The proposed model used an NS2 simulator and attained throughput of 90 rounds, energy consumption of 36mJ, delay of 27ms, packet delivery of 92%, and anomaly detection of 95%. However, the FIDS-HMM model does not improve the security level during the routing process. Aburasain, (2024) proposed an Enhanced Black Widow Optimization with Hybrid Deep Learning Enabled Intrusion Detection (EBWO-HDLID) technique. The proposed model attained 98.81% accuracy, 90.84% precision, 78.95% recall, and 79.49% F1-score for ToN-IoT dataset, and 98.35% accuracy, 84.85% precision, 80.95% recall, and 82.79% F1-score for Edge-IIoT dataset. However, the model's performance relies on the quality and diversity of the training data. Li & Yao, (2024) proposed a two-stage lightweight intrusion detection model based on self-supervised contrastive learning and self-knowledge distillation called CL-SKD. The model attained 99.95% accuracy on the NSL-KDD dataset, 99.80% on the CIC IDS2017 dataset, 100% on the BoT-IoT dataset, and 99.95% on the KDD CUP99 dataset. However, the developed model has high computational and requires high processing power and time.

3 Methodology

The proposed Chaotic Walrus Optimization-based Convolutional Echo State Networks (CWO-CESN) model is used to classify the attacks. The dataset used for IDS in WSN is collected from various repositories, including NSL-KDD, WSN-DS, and IoT-23. The missing values are removed from the collected data using the KNN method, and the data are normalized using the min-max normalization technique. The standard SMOTE method is used to solve the class imbalance issue. The processed data are given in the CESN model for feature learning and detection, in which the ESN model was integrated with the CNN model. The CNN model is used for capturing spatial features, and the ESN model is utilized to model temporal features to detect the attacks. The CESN model improves the performance of complex tasks by increasing classification and detection accuracy and produces more accurate solutions in WSN-based IOT networks. To further enhance the models' performance, the CWO technique was employed to optimize the hyperparameters of CESN. CWO integrated the Walrus Optimizer with chaos theory, which applied the chaotic theory concept to population initialization instead of randomization. Figure 1 represents the overall methods of the IDS model using CWO-CESN.

1) System Model

In the proposed IDS model for IoT-assisted WSN, the network comprises several nodes, including a source node, multicast senders, intermediate nodes, and destination nodes from the core structure of the

network. The source node originates the data that needs to be monitored in environmental conditions, and the collected data is transmitted through the network. The source node generates data, the multicast senders distribute it to multiple destinations, and the intermediate nodes facilitate communication across the network. The nodes are wirelessly interconnected with each node's transmission range. This network structure supports effective data collection, transmission, and intrusion detection within the WSN-IoT environment.

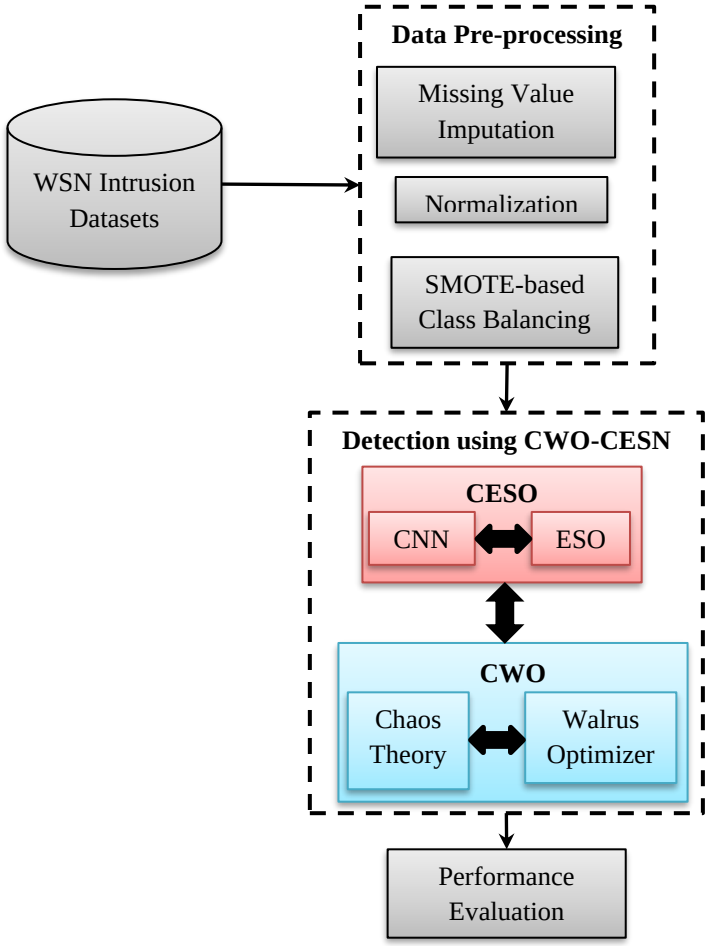

Figure 1: Proposed IDS Model Using CWO-CESN

2) Dataset Description

The research utilized three intrusion datasets to evaluate the performance of IIDS for WSN, namely NSL-KDD, WSN-DS, and IoT-23. The NSL-KDD dataset consists of two main classes, namely normal and attacks. The attacks are classified into DoS, Probe, R2L, and U2R attacks. NSL-KDD contains 41 features, including 9 basic features, 10 content features, 13 traffic features, and 1 feature of class label.

WSN-DS dataset consists of 12 attack classes, namely DoS, Sinkhole, Sybil, Wormhole, Black hole, Replay, Selective Forwarding, Spoofing, Hello Flood, Node Replication, Energy Exhaustion, and Routing Attacks. The features range from 20 to 40, typically including network traffic metrics, sensor data, and performance indicators.

IoT-23 is a synthetic dataset created to simulate network traffic involving IoT devices. It contains 81 features: basic network features, traffic and flow features, statistical features, protocol-specific features, and anomaly and attack Indicators. The attack types are classified into Botnet, DoS, DDoS, Port Scanning, Brute Force, SQL Injection, XSS, C2, and Normal Traffic. Characteristics of the Datasets shown in table 1.

Table 1: Characteristics of the Datasets

DATASET	ATTACK CLASS	DATA CLASS
NSL-KDD	Dos, U2R, Probe Attacks	Basic, Content, and Traffic
WSN-DS	Black hole, Flooding, DoS attacks in WSN, Gray hole, and Scheduling attacks in IoT.	Normal, Gray hole
IoT-23	Twenty network attacks from infected IoT devices and three networks from real IoT devices	Benign and Malicious

3) Data Pre-processing

The collected data are pre-processed to improve the model's performance effectively. The missing values are removed and replaced using the KNN technique, which considers the point nearest to the data value. The nearest value (K) is found, and using the Euclidean distance formula, the distance between the instances and the missing values is calculated. The equation (1) is given as,

$$E_D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Here, $x_i \; and \; y_i$ denotes the data values and $E_D(x, y)$ represents the Euclidean distance between the data points.

The nearest values (K) are selected based on the shortest distance between the missing value and the instance. To replace the missing values, the obtained nearest values are computed using the median imputation method, and it is formulated the equation (2) is given as,

$$\hat{X}_{ij} = median(x_1, x_2, \dots, x_k) \quad (2)$$

Here, $x_1, x_2, \dots, x_k$ represents K nearest neighbor indices.

After replacing the missing values, the min-max technique is employed for data normalization. The min-max technique transforms the data based on the fixed range with [0,1]. This process is formulated the equation (3) is given as,

$$\hat{x} = y + \frac{(x - x_{min})(z - y)}{x_{max} - x_{min}} \quad (3)$$

Here, $\tilde{a}$ refers to the normalized value, $x_{max} \; and \; x_{min}$ represents the maximum and minimum value, and $y \; and \; z$ refers to the scaling factors that range between [0,1].

After data is normalized, the SMOTE method resolves the class imbalance issues. SMOTE enhances the overlapping of classes in the normalized data by generating synthetic samples for the majority classes, which balance the class distribution. The SMOTE selects a K-nearest neighbor point for each majority class instance. The synthetic values are generated using the KNN, and a new data value is generated along with the line segment. The difference is calculated between the majority class instances, (x_i) and the nearest neighbor $(x_{i,j})$ are formulated the equation (4) is given as,

$$diff = x_{i,j} - x_i \quad (4)$$

Here, x_i refers to the majority class instance and $x_{i,j}$ refers to the nearest neighbor. The random number is generated between the range 0 and 1 and is formulated the equation (5) is given as,

$$interval = \vartheta \times diff \quad (5)$$

The synthetic sample $\left(x_{synthetic}\right)$ is generated by utilizing the interval instance value with the original class instance value, and it is computed. The equation (6) is given as,

$$x_{new} = x_i + interval \quad (6)$$

Here, x_{new} refers to the generated point that lies on the line segment between $x_i \; and \; x_{i,j}$ and $interval$ refers to the random number.

4) CESN Model for Intrusion Detection

The processed data are given to the CWO-CESN model to classify the attacks. CESN model is developed by integrating Convolutional Neural Networks with Echo State Networks. The CNN model is a Neural Network type incorporating convolutional layers, automatically extracting the input data's characteristics. The CNN model contains four components: convolutional layers, pooling layers, and flattening layers. The features are analyzed and extracted through the convolution layer (CL). The convolution operation retains the spatial relationship between the input data by capturing the features through the kernel function. This process is formulated the equation (7) is given as,

$$Y_{ij} = (x * W)_{ij} + b \quad (7)$$

Here, x refers to the processed data, W refers to the weight, and b refers to the bias term.

The ReLU activation function is also employed to improve computing efficiency. The calculation process of the convolutional layer and the equation (8) is given as,

$$f = \varphi(\omega^n \oplus x + b^n) \quad (8)$$

Here, ω represents weighing factors in kernels, n represents the number of kernels in convolutional layers, x represents the vector of input series, and b represents bias. $\oplus$ indicates convolutional operation, and φ denotes the ReLU activation function.

After the convolutional operation, the obtained features have large dimensions, so a pooling layer is applied to reduce the number of dimensions efficiently, performing maximum pooling operation. The equation (9) is given as,

$$Y_P = Pool_{max}(Y_c) \quad (9)$$

Here, Y_P represents the output of pooling layers and $Pool_{max}$ represents the maximum pooling function. After the pooling layers, the multi-dimensional feature maps are transformed into a one-dimensional array using a flattened layer.

After the flattening layer ESN model is integrated, the hidden RNN layers are replaced by the reservoir layer to improve the efficiency of computations. It comprises the input layer, reservoir layer, and output layer. The weights of both input and reservoir are randomly initialized, and the reservoir layer is initialized with very sparse connections to encourage multiple oscillatory dynamics. The output weights are calculated using linear regression. The N-dimensional initial states are given as x (0), and the D-dimensional input series are provided as $U = (u(1), \ldots, u(t), \ldots, u(T))^T$ and L-dimensional output series is given as $Y = (y(1), \ldots, y(t), \ldots, y(T))^T$. The ESN model has three essential properties: temporal kernel, echo state property (ESP), and high training efficiency. The temporal kernel represents the reservoir layer's ability to change the input sequence into an echo state response with

high–dimensionality, a nonlinear representation of the input series data. This process enables the ESN to gather the complex temporal dependencies. The reservoir transformation and the equation (10) is given as,

$$x(t) = f(W^{res}\, x(t-1) + W^{in}\, u(t)) \qquad (10)$$

Here. $u\,(t) \in \mathbb{R}^{D}$ refers to the vector of D-dimension at time t, $x(t) \in \mathbb{R}^{L}$ refers to the vector of L-dimension of the reservoir at time t, W^{res} denotes the connection weight within the reservoir, W^{in} denotes connection weights from the input layer to the reservoir, and f denotes the $tanh$ activation function.

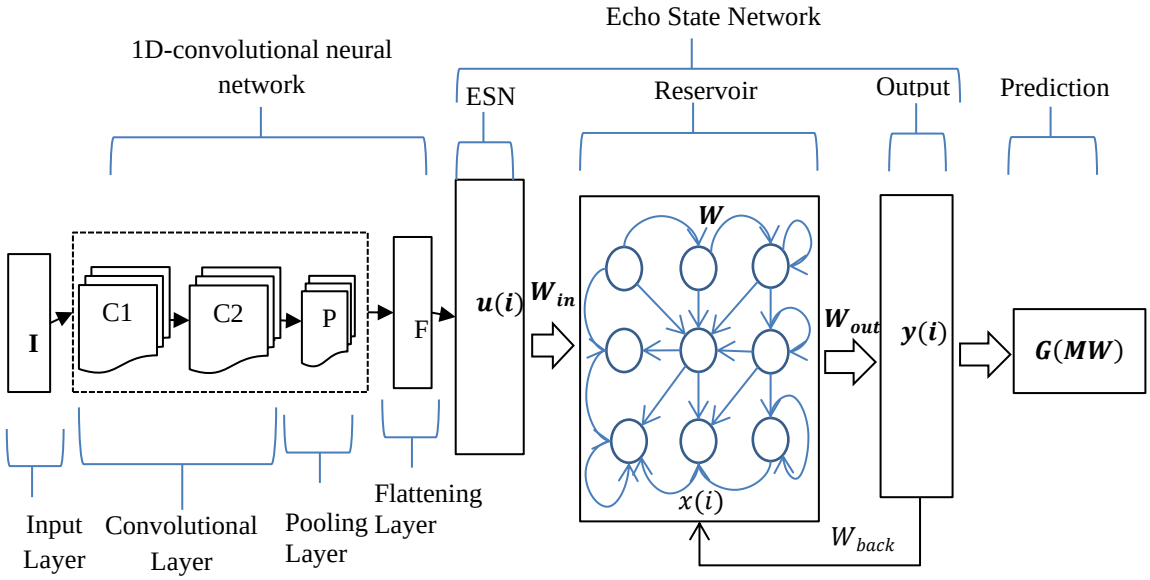

Figure 2: Architecture of CESN Model

Figure 2 represents the construction of the CESN model, which integrates an ESN with a CNN to form the CESN model, designed for predicting electricity demand (G) in a day. In this framework, the CNN component is responsible for feature extraction from the lagged series of G, capturing both common and local trends over varying periods. From CNN layers, features are extracted and then flattened before being given to ESN, which acts as a predictive regression operator for the G data. The ReLU activation function is employed in the CNN, while the Adam algorithm is utilized for back propagation, with the fully connected layer replacing ESN in their CESN model.

ESP ensures the current state of the reservoir is the function of the recent input history. It makes a reservoir to prevent the earlier input series history information from fading. The ESP condition is formulated the equation (11) is given as,

$$\lim_{t\to\infty} x(t) = x^{final} \qquad (11)$$

Here, the influence of the initial state $x(0)$ on the reservoir state $x(t)$diminishes over time, and the state $x(t)$ is given by the recent input sequence $u(t)$.

The high training efficiency of ESN comes from the output weights. W^{out} to be trained through linear regression. The output is calculated and the equation (12) is given as,

$$y(t) = f^{out}(W^{out}x(t)) \qquad (12)$$

Here, W^{out} denotes the weight connection from the reservoir to the output layer, and f^{out} denotes $tanh$ activation function of the output layer.

CESN has combined the feature extraction from CNN with the temporal processing of ESN, with its high training efficiency and adaptability for IDS in IoT-assisted WSNs. To further increase the performance of the CESN model, the hyperparameters are fine-tuned by using CWO. The CWO model integrates the advantages of Chaos theory and Walrus optimizer (WO). The Chaotic maps from the chaos theory are used for population initialization, and the initiated population is fed to the Walrus optimizer to obtain the optimal weight and bias.

The optimization begins with the candidate solution set (X), which is generated randomly and the equation (13) is given as,

$$X = LB + rand(UB - LB) \quad (13)$$

Here, LB refers to the lower boundary of the variables, and UB refers to the upper boundary of the variable.

WO consists of randomly constructed models that are replaced by chaotic maps during the initialization of the population. These maps effectively improve the process of exploitation and exploration. To avoid local optimum and increase diversity, this map provides the concept of unpredictability and randomness. The efficiency of these maps purely depends on the global exploration of the search area, which shows their effectiveness.

The logical map is chosen as a chaotic map for this proposed model that serves as a metaphor for dynamic evolution. Logical maps are used to learn chaotic systems' complex behaviour and relationships. This logical map is used to understand and analyze the intricate behaviour of chaotic systems and produce insights for complicated relationships and mapping within the systems. This logical map is formulated and the equation (14) is given as,

$$x_{n+1} = ax_n(1 - x_n) \quad (14)$$

Here, a denotes a parameter for defining the mapping behaviour, n represents the number of iterations, x_n represents chaotic numbers and x_{n+1} represents the state of the next iteration. Therefore, equation (15) is rewritten as,

$$q_{j,k} = l_k + S_l(u_k - l_k), j - 1,2,3, \dots, G, k = 1,2,3, \dots, m \quad (15)$$

Where, S_l denotes l^{th} iteration chaotic sequence output. The chaotic map is used to initialize the positions to improve global search performance in the WO algorithm.

Walruses are highly vigilant during foraging and roosting, with one or two acting as guards who patrol the area. If any unexpected situation arises, they promptly send out danger signals. In WO, these danger and safety signals are defined as follows. The equation (16-19) is given as,

$$Danger_{signal} = A * R \quad (16)$$
$$\alpha = 1 - c/T \quad (17)$$
$$A = 2 \times a \quad (18)$$
$$R = 2 \times r_1 - 1 \quad (19)$$

Here, Danger factors are represented by $A\ and\ R$, α reduces from 1 to 0 within the iterations count c, and T is the highest iteration.

The safety signal in WO that reflects the danger signal is written and the equation (20) is given as,

$$Safety_{signal} = r_2 \quad (20)$$

Here, $r_1 \; and \; r_2$ represents the random numbers that range between $0 \; and \; 1$.

The exploration phase is represented as the migration process. During high-risk factors, walrus herds relocate for survival. During the exploration phase, the positions of the walruses are updated as follows and the equation (21-23) is given as,

$$X_{i,j}^{c+1} = X_{i,j}^{c} + Migration_{step} \quad (21)$$

$$Migration_{step} = (X_m^c - X_n^c) \cdot \beta \cdot r_3^2 \quad (22)$$

$$\beta = 1 - \frac{1}{1+\exp\left(-\frac{c-\frac{T}{2}}{T}\times 10\right)} \quad (23)$$

Here, $X_{i,j}^{c+1}$ refers to the next position on the j^{th} dimension by the i^{th} walrus, $X_{i,j}^{c}$ refers to the current position, $Migration_{step}$ represents the size of the walrus movement step, $X_m^c \; and \; X_n^c$ refers to the vigilantes, β represents the control factor for the migration step, and r_3 represents the random variable between $0 \; and \; 1$.

The exploitation phase is represented as the reproduction of walrus herds. The walrus herds start breeding in low-risk environments. During reproduction, two main behaviors are analyzed: onshore roosting and underwater foraging. In Roosting behavior, the walrus population is divided into males, females, and juveniles, each having unique ways of updating their positions, including male walrus (MW) redistribution, Female walrus (FW) Position update, and Position update of juvenile walruses (JW). The male walrus (MW) redistribution uses the Halton sequence in the quasi-Monte Carlo method to update the position of the male walrus to ensure an evenly distributed population.

The FW Position update is impacted by both a MW $(Male_i, j)$ and the lead walrus (X_{best}^t). As iterations progress, the influence of the male diminishes while the leader's influence grows more robust. The equation (24) is given as,

$$Fe_{i,j}^{c+1} = Fe_{i,j}^{c} + \alpha \cdot \left(M_{i,j}^{c} - Fe_{i,j}^{c}\right) + (1-\alpha) \cdot (X_{best}^{c} - Fe_{i,j}^{c}) \quad (24)$$

Here, $Fe_{i,j}^{c+1}$ refers to the new position for the i^{th} female walrus, $M_{i,j}^{c}$ and $Fe_{i,j}^{c}$ refers to the positions of the MW and FW on the j^{th} dimension.

The juvenile walruses are in the edge population, vulnerable to predators such as killer whales and polar bears. As a result, they must frequently update their positions to evade potential threats. The equation (25-26) is given as,

$$Juvenile_{i,j}^{c+1} = (O - Juvenile_{i,j}^{c}) \cdot P \quad (25)$$

$$O = X_{best}^{c} + Juvenile_{i,j}^{c} \cdot LF \quad (26)$$

Here, $Juvenile_{i,j}^{c+1}$ refers to the new position, $Juvenile_{i,j}^{c}$ refers to the current position, O refers to the safety position, P refers to the random number between $(0,1)$ and also distress coefficient and LF refers to the levy distribution random number. The levy movement is represented and the equation (27) is given as,

$$Levy(a) = 0.05 \times \frac{x}{|y|^{\frac{1}{a}}} \quad (27)$$

Here, $x \; and \; y$ refers to the distributed variable, and it is represented as $x = N\left(0, \sigma_x^2\right)$ and $y = N\left(0, \sigma_y^2\right)$. The equation (28) is given as,

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\frac{\Pi\alpha}{2})}{\Gamma(\frac{1+\alpha}{2})\alpha 2^{\frac{(\alpha-1)}{2}}}\right]^{\frac{1}{\alpha}}, \sigma_y = 1, \alpha = 1.5 \quad (28)$$

Here, σ_x and σ_y are the representation of standard deviations.

The Foraging behavior in the exploitation phase includes fleeing behavior and gathering behavior.

The fleeing behavior represents the attacks walruses face during underwater foraging from natural predators. In response to danger signals from their peers, they quickly flee their current location. This behavior typically occurs in the later iterations of the Walrus optimization process, where introducing some perturbation to the population aids in global exploration. The equation (29) is given as,

$$X_{i,j}^{c+1} = X_{i,j}^{c} \cdot R - |X_{best}^{c} - X_{i.j}^{c}| \cdot r_4^2 \quad (29)$$

Here, $|X_{best}^{c} - X_{i.j}^{c}|$represents the difference between the present walrus and the best walrus, r_4 shows a random variable that lies in the range between 0 and 1.

When engaging in gathering behavior, walruses work together to find food, move in with the other walruses in the group, and provide position data that helps the herd determine the sea area with high food availability. This process and the equation (30-34) is given as,

$$X_{i,j}^{c+1} = (X_1 + X_2)/2 \quad (30)$$
$$X_1 = X_{best}^{c} - a_1 \times b_1 \times |X_{best}^{c} - X_{i.j}^{c}| \quad (31)$$
$$X_2 = X_{second}^{c} - a_2 \times b_2 \times |X_{second}^{c} - X_{i.j}^{c}| \quad (32)$$
$$a = \beta \times r_5 - \beta \quad (33)$$
$$b = \tan(\theta) \quad (34)$$

Here, $X_1 and\ X_2$ are two weights that affect the gathering behavior of walruses, X_{second}^{t} refers to the position which shows other walrus the present iterations, $a\ and\ b$ refers to the coefficients of gathering, r_5 refers to the random variable that ranges between 0 and 1.

Algorithm 1. CWO-CESN

1. The parameters are initialized
2. The population is initialized by using a chaotic map
3. The fitness value is calculated for each solution
4. While $c \leq T$
5. // Exploration Phase (Migration Process)
    a. For each walrus i in the population, DO:
        i. If danger signal detected:
            1. Update walrus position:
        ii. End if
6. // Exploitation Phase (Reproduction Process)
        i. If the walrus is male:
            1. Redistribute male walruses using the Halton sequence
            2. Update male walrus position based on uniform distribution
        ii. Else if the walrus is female:
            1. Update female walrus position influenced by male and leader
        iii. Else, if the walrus is juvenile, then:
            1. Update juvenile walrus position to evade predators
        iv. End if

a. End for
7. //Foraging Behavior in the Exploitation Phase
    i. If a danger signal is detected during foraging:
        1. Fleeing behavior to avoid predators
    ii. Else
    iii. Gathering behavior for cooperation during foraging
    iv. End if
    a. End for
    b. Update the best solution based on fitness evaluation
8. End While
9. The optimal solution and weight and bias are returned
10. End

4 Performance Evaluation

The CWO-CESN model is evaluated, and the performance is analyzed using metrics, including accuracy, precision, recall, F-score, and processing time (PT). The proposed CWO-CESN is compared with the existing models, and the performance is analyzed for the three datasets, including NSL-KDD, WSN-DS, and IoT-23 datasets.

Table 2: Comparison of CWO-CESN with the Existing Models for the NSL-KDD Dataset

Methods	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	FPR (%)	FNR (%)
ECRF [12]	NSL-KDD	97.17	97.32	97.02	95.92	14.8	15.75
CNN [17]	NSL-KDD	97.01	97	94.09	95.50	11.2	11.81
FA-ML [19]	NSL-KDD	96.42	94.40	95.25	95.42	15.3	10.11
CL-SKD [25]	NSL-KDD	97.55	97.50	97.55	97.49	10.7	9.76
CWO-CESN	NSL-KDD	99	98.89	98.77	98.82	6.51	7.7

Table 2 demonstrates the comparison results with the existing methods for the NSL-KDD dataset. The CWO-CESN model increased accuracy by 1.83%, 1.99%, 2.58%, and 1.45%, precision increased by 1.57%, 1.89%, 4.49%, and 1.39%, recall increased by 1.75%, 4.68%, 3.52%, and 1.22%, and F1-score increased by 2.9%, 3.32%, 3.4%, and 1.33%, increased FPR by 8.29%, 4.69%, 8.79%, and 4.19%, increased FNR by 8.05%, 4.11%, 2.41%, and 2.06% for ECRF, CNN, FA-ML, and CL-SKD models.

Table 3: Comparison of CWO-CESN with the Existing Models for the WSN-DS Dataset

Methods	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPR (%)	FNR (%)
BCOA-MLID [9]	WSN-DS	97.91	94.52	94.54	94.52	14.95	10.4
Bi-GRU-AE [13]	WSN-DS	99.34	99.34	99.34	99.34	15.67	13.8
FL-SCNN-Bi-LSTM [20]	WSN-DS	98.7	98.8	98.6	98.6	13.7	11.53
ML-SMOTE-TomekLink [21]	WSN-DS	98.80	98.81	98.79	98.8	11.67	8.82
CWO-CESN	WSN-DS	99.5	99.6	99.6	99.6	7.54	6.53

Table 3 compares the results with the existing methods for the WSN-DS dataset. The CWO-CESN increased accuracy by1.59%, 0.7%, 0.8%, and 0.16%, precision increased by, 5.08%, 0.79%, 0.8%, and 0.26% recall increased by 5.08%, 0.81%, 1%, and 0.26%, and F1-score increased by 5.08%, 0.8%, 1%,

and 0.26%, increased FPR by 7.41%, 4.13%, 6.16%, and 8.13%, increased FNR by 3.87%, 2.29%, 5%, and 7.25% BCOA-MLID, ML-SMOTE-TomekLink, FL-SCNN-Bi-LSTM, and Bi-GRU-AE models.

Table 4: Comparison of CWO-CESN with the Existing Models for the IoT-23 Dataset

Methods	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPR (%)	FNR (%)
CNN-LSTM [150]	IoT-23	98.80	98.67	98.5	98.6	10.6	13.9
1D-CNN-LSTM [22]	IoT-23	98.75	98.75	98.75	98.75	15.95	17.3
Ensemble model [16]	IoT-23	98.9	98.83	98.68	98.02	13.5	15.7
EBWO-HDLID [24]	IoT-23	98.35	84.85	80.95	82.79	12.4	14.4
CWO-CESN	IoT-23	99.8	99.6	99.7	99.6	8.4	9.2

Table 4 compares the results with the existing methods for the IoT-23 dataset. The CWO-CESN increased accuracy by 1.05%, 0.9%, 1.45%, and 1%, precision increased by 0.85%,0.77%, 0.93, recall increased by 0.85%, 1.02%, 18.75%, and 1.2%, and F1-score increased by 0.85%, 1.58%, 16.84%, and 1% for 1D-CNN-LSTM, Ensemble model, EBWO-HDLID, and CNN-LSTM models.

Table 5: Processing Time Comparison

Dataset	No-of samples	Training samples	Testing samples	No of features	Processing time (s)
NSL-KDD	125973	88181	37792	42	97.12
WSN-DS	374661	262262	112311	19	137.45
IoT-23	48003	33602	14401	58	45.02

Table 5 represents the processing time for the NSL-KDD, WSN-DS, and IoT-23 datasets, along with the total samples, training samples, testing samples, and several features. The results showed that the proposed CWO-CESN model significantly reduces the overall processing time for all three datasets. The accuracy, precision, recall, F-score, FPR, and FNR values also show that the model has higher detection rates than the compared models. The utilization of the hybrid DL approach of CESN has enhanced the detection rate by learning the deep feature patterns, and the processing time is also reduced by the training optimization of CESN using the CWO algorithm.

5 Conclusion

The developed CWO-CESN model significantly improves the Intrusion detection system (IDS) for WSN and IoT networks, addressing the critical challenges in network security. The model employs a process that includes a data pre-processing phase, utilizing KNN imputation for missing values and Min-Max normalization for scaling. At the same time, the SMOTE method reduces class imbalance issues, ensuring high-quality data for intrusion detection. The hybrid CESN model efficiently gathers network traffic's spatial and temporal dynamics and allows for detecting complex patterns and anomalies. The CNN component excels in pattern recognition, while ESN efficiently models temporal dependencies with computational efficiency. The CWO algorithm optimizes the hyperparameters of the CESN model, leveraging the strengths of chaos theory and the Walrus optimizer to enhance detection accuracy. This

hyperparameter tuning process enhances the model's generalization ability on different attacks. The proposed CWO-CESN model has been evaluated by utilizing benchmark datasets, NSL-KDD, WSN-DS, and IoT-23, attaining accuracy rates of 99%, 99.5%, and 99.8%, respectively, for detecting various attacks. The results demonstrate the model's robustness and potential for real-time intrusion detection in WSN and IoT networks. The possibility of including more new attack patterns will be examined in the future. Additionally, the feasibility of further reducing the training time will be investigated.

References

[1] Aburasain, R. Y. (2024). Enhanced Black Widow Optimization with Hybrid Deep Learning Enabled Intrusion Detection in Internet of Things-based Smart Farming. *IEEE Access*, 12, 16621-16631. https://doi.org/10.1109/ACCESS.2024.3359043

[2] Adnan, A., Muhammed, A., Abd Ghani, A. A., Abdullah, A., & Hakim, F. (2021). An intrusion detection system for the Internet of things based on machine learning: Review and challenges. *Symmetry, 13*(6), 1011. https://doi.org/10.3390/sym13061011

[3] Al Sawafi, Y., Touzene, A., & Hedjam, R. (2023). Hybrid deep learning-based intrusion detection system for RPL IoT networks. *Journal of Sensor and Actuator Networks, 12*(2), 21. https://doi.org/10.3390/jsan12020021

[4] Aljebreen, M., Alohali, M. A., Saeed, M. K., Mohsen, H., Al Duhayyim, M., Abdelmageed, A. A., & Abdelbagi, S. (2023). Binary chimp optimization algorithm with ML-based intrusion detection for secure IoT-assisted wireless sensor networks. *Sensors, 23*(8), 4073. https://doi.org/10.3390/s23084073

[5] Alruwaili, F. F., Asiri, M. M., Alrayes, F. S., Aljameel, S. S., Salama, A. S., & Hilal, A. M. (2023). Red kite optimization algorithm with average ensemble model for intrusion detection for secure IoT. *IEEE Access, 11*, 131749-131758.

[6] Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+ LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, *38*, 101322. https://doi.org/10.1016/j.jestch.2022.101322.

[7] Awotunde, J. B., Folorunso, S. O., Imoize, A. L., Odunuga, J. O., Lee, C. C., Li, C. T., & Do, D. T. (2023). An ensemble tree-based model for intrusion detection in industrial Internet of things networks. *Applied Sciences, 13*(4), 2479. https://doi.org/10.3390/app13042479

[8] Ba Hmaid, S. A. A., & Varadharajan, V. (2020). Multipath Data Transmission in IoT Networks Using Fractional Firefly Algorithm and Chicken Swarm Optimization. *International Journal of Intelligent Engineering & Systems*, *13*(3), 373-383.

[9] Ba Hmaid, S. A. A., & Vasanthi, V. (2020). Fractional Gaussian firefly algorithm and Darwinian chicken swarm optimization for IoT multipath fault-tolerant routing. *International Journal of Computer Networks and Applications, 7*(6), 167-177.

[10] Biswas, P., Samanta, T., & Sanyal, J. (2023). Intrusion detection using graph neural network and Lyapunov optimization in wireless sensor network. *Multimedia Tools and Applications*, *82*(9), 14123-14134.

[11] Bukhari, S. M. S., Zafar, M. H., Abou Houran, M., Moosavi, S. K. R., Mansoor, M., Muaaz, M., & Sanfilippo, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Networks, 155*, 103407. https://doi.org/10.1016/j.adhoc.2024.103407

[12] Camgözlü, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences, 8*(3), 214-232.

[13] Ghaleb, S. A. M., & Varadharajan, V. (2020). Convergence Factor and Position Updating Improved Grey Wolf Optimization for Multi-constraint and Multipath QoS Aware Routing in

Mobile Adhoc Networks. *International Journal of Intelligent Engineering and Systems, 13*(4), 457-466.

[14] Hui, H., An, X., Wang, H., Ju, W., Yang, H., Gao, H., & Lin, F. (2019). Survey on Blockchain for Internet of Things. *Journal of Internet Services and Information Security, 9*(2), 1-30.

[15] Jelena, T., & Srđan, K. (2023). Smart Mining: Joint Model for Parametrization of Coal Excavation Process Based on Artificial Neural Networks. *Archives for Technical Sciences, 2*(29), 11-22.

[16] Karthic, S., & Kumar, S. M. (2023). Hybrid optimized deep neural network with enhanced conditional random field-based intrusion detection on wireless sensor network. *Neural Processing Letters, 55*(1), 459-479.

[17] Karthikeyan, M., Manimegalai, D., & Raja Gopal, K. (2024). Firefly algorithm-based WSN-IoT security enhancement with machine learning for intrusion detection. *Scientific Reports, 14*(1), 231. https://doi.org/10.1038/s41598-023-50554-x

[18] Khedr, A. M., Osamy, W., Salim, A., & Abbas, S. (2020). A novel association rule-based data mining approach for Internet of things based wireless sensor networks. *IEEE Access, 8*, 151574-151588.

[19] Krishnan, R., Krishnan, R. S., Robinson, Y. H., Julie, E. G., Long, H. V., Sangeetha, A., & Kumar, R. (2022). An intrusion detection and prevention protocol for Internet of Things-based wireless sensor networks. *Wireless Personal Communications, 124*(4), 3461-3483.

[20] Li, Z., & Yao, W. (2024). A two stage lightweight approach for intrusion detection in Internet of Things. *Expert Systems with Applications*, *257*, 124965. https://doi.org/10.1016/j.eswa.2024.124965.

[21] Narayanan, S. L., Kasiselvanathan, M., Gurumoorthy, K. B., & Kiruthika, V. (2023). Particle swarm optimization-based artificial neural network (PSO-ANN) model for effective k-barrier count intrusion detection system in WSN. *Measurement: Sensors*, *29*, 100875. https://doi.org/10.1016/j.measen.2023.100875

[22] Ram, A., & Chakraborty, S. K. (2024). Analysis of Software-Defined Networking (SDN) Performance in Wired and Wireless Networks Across Various Topologies, Including Single, Linear, and Tree Structures. *Indian Journal of Information Sources and Services, 14*(1), 39–50.

[23] Ramana, K., Revathi, A., Gayathri, A., Jhaveri, R. H., Narayana, C. L., & Kumar, B. N. (2022). WOGRU-IDS—An intelligent intrusion detection system for IoT-assisted Wireless Sensor Networks. *Computer Communications, 196*, 195-206.

[24] Ravindran, S. (2024). Intelligent fuzzy logic based intrusion detection system for effective detection of black hole attack in WSN. *Peer-to-Peer Networking and Applications*, 1-17.

[25] Skarmeta, A.F., Cano, M.V.M., & Iera, A. (2015). Guest Editorial: Smart Things, Big Data Technology and Ubiquitous Computing solutions for the future Internet of Things. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 6*(1), 1-3.

[26] Subramani, S., & Selvi, M. (2023). Multi-objective PSO-based feature selection for intrusion detection in IoT-based wireless sensor networks. *Optik, 273*, 170419.

[27] Talukder, M. A., Sharmin, S., Uddin, M. A., Islam, M. M., & Aryal, S. (2024). MLSTL-WSN: machine learning-based intrusion detection using SMOTE Tomek in WSNs. *International Journal of Information Security, 23*(3), 2139-2158.

[28] Vembu, G., & Ramasamy, D. (2023). Optimized deep learning-based intrusion detection for wireless sensor networks. *International Journal of Communication Systems, 36*(13), e5254. https://doi.org/10.1002/dac.5254

[29] Yao, W., Hu, L., Hou, Y., & Li, X. (2023). A lightweight intelligent network intrusion detection system using one-class autoencoder and ensemble learning for IoT. *Sensors, 23*(8), 4141. https://doi.org/10.3390/s23084141

[30] Yaras, S., & Dener, M. (2024). IoT-based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. *Electronics, 13*(6), 1053. https://doi.org/10.3390/electronics13061053

Authors Biography

Dr. Salem Bahmaid, received his Bachelor of Computer Applications (BCA) in 2014 from Mysore University, Mysore, India, and his Master of Information Technology (M.IT) in 2016 from Bharathiar University, Coimbatore, India. He was awarded a Ph.D. in Networking within the field of Information Technology in 2022 from Bharathiar University, Coimbatore, India. Currently, he holds the position of Assistant Professor at the Arab Open University in Riyadh, Saudi Arabia. His research interests include the Internet of Things (IoT), Advanced Networking, Cybersecurity, and Artificial Intelligence (AI).

Dr. Sulaiman Abdo Mahyoub Ghaleb, obtained his Bachelor of Computer Applications (BCA) in 2014 from Mysore University, Mysore, India, and his Master of Information Technology (M.IT) in 2016 from Bharathiar University, Coimbatore, India. He was awarded a Ph.D. in Networking within the field of Information Technology in 2022 from Bharathiar University, Coimbatore, India. He is an Assistant Professor at Arab Open University, Riyadh, Saudi Arabia. His research interests include Advanced Networking, Cybersecurity, Artificial Intelligence, Cloud Computing, and the Internet of Things (IoT).