scientific reports

Check for updates

OPEN

Leveraging hybrid deep learning with starfish optimization algorithm based secure mechanism for intelligent edge computing in smart cities environment

Amal K. Alkhalifa[1], Mohammed Aljebreen[2], Rakan Alanazi[3]✉, Nazir Ahmad[4], Othman Alrusaini[5], Nojood O. Aljehane[6], Ali Alqazzaz[7] & Hassan Alkhiri[8]

The Internet of Things (IoT) now appears in each domain, from smart cities to home applications. The widespread use of IoT is making its security a real concern. The past few years have revealed an extraordinary increase in computer-intensive applications. Such applications always make huge volumes of data that demand severe latency-aware computational processing abilities. While edge computing is one of the attractive technologies for balancing severe latency-related problems, its deployment produces novel tasks. Edge computing is an innovative model distinguished mainly by its mobility support, geo-distributed process, low latency, and context awareness. However, recent edge computing developments have begun to explore novel IoT potentials that are leveraged from a security perspective. Methods depend upon artificial intelligence (AI) and its subgroups, machine learning (ML) and deep learning (DL), are generally employed to develop a safe Intrusion Detection System (IDS) for IoT. This study proposes a Hybrid Deep Learning-Based Intrusion Detection for Edge Computing Using Starfish Optimization Algorithm (HDLID-ECSOA) technique. The main goal of the HDLID-ECSOA technique is to provide intelligent edge computing in smart cities using advanced optimization models. Initially, the data pre-processing employs the min-max normalization to convert and standardize raw data to improve the efficiency of models. Furthermore, the dingo optimizer algorithm (DOA) technique detects and chooses the most relevant features from input data. Moreover, integrating a convolutional neural network and bidirectional gated recurrent unit with a cross-attention mechanism (CNN-BiGRU-CrAM) technique is implemented for the classification process. To enhance model performance, the starfish optimization algorithm (SFOA) is used for hyperparameter tuning to select the optimal parameters for improved accuracy. A comprehensive experimentation analysis of the HDLID-ECSOA model is performed under the Edge-IIoT and ToN-IoT datasets. The experimental validation of the HDLID-ECSOA model portrayed superior accuracy values of 99.35% and 99.33% over existing techniques under the dual dataset.

Keywords Intrusion detection , Edge computing, Deep learning, Starfish optimization algorithm, Dingo optimizer algorithm

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. [2]Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia. [3]Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia. [4]Department of Computer Science, Applied College at Mahayil, King Khalid University, King Khalid, Saudi Arabia. [5]Department of Engineering and Applied Sciences, Applied College, Umm Al-Qura University, Makkah, Saudi Arabia. [6]Department of Computer Science, Faculty of Computers and Information Technology, University of Tabuk, Tabuk, Saudi Arabia. [7]Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Bisha 67714, Saudi Arabia. [8]Department of Computer Science, Faculty of Computing and Information Technology, Al-Baha University, Saudi ArabiaAl-Baha. ✉email: rakan.nalenezi@nbu.edu.sa

Smart cities are metropolitan regions utilizing data-driven technology to enhance residents' sustainability, efficiency, and living standards[1]. The concept of smart cities has recently attained substantial traction owing to the progression of IoT, AI, and big data. It has experienced a phenomenal increase in domain-specific applications, namely smart agriculture, healthcare, industry, and smart transportation systems, to enhance socio-economic improvement in recent times[2]. These IoT methods are formed by various actuators, interconnected sensors, and network-enabled gadgets that can interchange diverse kinds of information through either Internet infrastructure or private networks[3]. The evolution of IoT gadgets has also improved the data system bandwidth. Nevertheless, IoT gadgets have constrained resources, making it challenging to perform the conventional security models for protecting systems against cyber-threats[4]. Thus, it is vital to present the multi-access edge computing (MEC), which permits the computation to be conducted at the edge of a system for tackling the resource-constrained issue in IoT networks. Figure 1 illustrates the architecture of edge computing in smart cities.

Edge computing is an effective method of refining the execution of machines and gadgets, tackling the drawback of cloud computing (CC)[5]. Edge computing is a structure that combines storage, network services, and computing, which extends from CC to the edges of the network. In comparison with the services of CC, edge computing increases computing speed, reduces storage, improves data security, lowers latency bandwidth, and decreases location limitations by offloading certain computations to edge devices[6]. In edge computing, multiple gadgets will create vast amounts of data; edge computing offers effective, secure services for several end-users. IoT has turned the driving force of the existing industrial revolution and the method to gather live dependent information, making it crucial to take cybersecurity seriously[7]. Thus, there is a requirement for an IDS which can identify existing and upcoming threats to safeguard the IoT systems and networks made by it. The IDS for edge computing settings identifies intruders using signature-based and anomaly-based methods. The normal behaviour of this method is anomaly-based recognition that inspects the behaviour of incoming traffic and classifies it as both abnormal and normal depending on the constructed technique. Conversely, signature-based recognition relates incoming traffic to pre-determined guidelines[8]. Recently, various investigation report articles have advanced in the field of IDS for edge computing (EC) settings. Earlier investigations focused on DL and ML methodologies[9]. There have also been endeavours to apply sophisticated applications, namely a traditional recognition technique that permits the integration of outcomes of multiple classifications to enhance the performance of IDS effectively. Traditional ML models are unsuitable for utilizing large volumes of data, owing to the absence of annotated trained data and the higher prominence on recovered features gained by users[10]. DL, an innovative technology in ML, employs artificial neural networks (ANN) and exceeds classical models.

This study proposes a Hybrid Deep Learning-Based Intrusion Detection for Edge Computing Using Starfish Optimization Algorithm (HDLID-ECSOA) technique. The main goal of the HDLID-ECSOA technique is to provide intelligent EC in smart cities using advanced optimization models. Initially, the data pre-processing employs the min-max normalization to convert and standardize raw data to improve the efficiency of models. Furthermore, the dingo optimizer algorithm (DOA) technique detects and chooses the most relevant features from input data. Moreover, integrating a convolutional neural network and a bidirectional gated recurrent unit with cross-attention mechanism (CNN-BiGRU-CrAM) technique is implemented for the classification process. To enhance model performance, the starfish optimization algorithm (SFOA) is used for hyperparameter tuning to select the optimal parameters for improved accuracy. A comprehensive experimentation analysis of the HDLID-ECSOA model is performed under the Edge-IIoT and ToN-IoT datasets. The key contribution of the HDLID-ECSOA model is listed below.

- The HDLID-ECSOA method applies min-max normalization to scale input features within a consistent range, ensuring balanced input for the learning process. This improves training stability and accelerates convergence. It also mitigates the risk of bias from dominant features, improving the model's more accurate and reliable performance.
- The HDLID-ECSOA technique utilizes the DOA method to effectually select the most relevant features from the dataset, eliminating redundant and irrelevant data. This enhances learning efficiency and mitigates overfitting risks. By reducing feature space, it also improves computation and model generalization.
- The HDLID-ECSOA approach effectively captures spatial and sequential patterns by combining a hybrid CNN-BiGRU technique with the CrAM model. This incorporation strengthens feature representation and context understanding. Concentrating on the most informative features significantly enhances classification accuracy, particularly in intrinsic data scenarios.
- The HDLID-ECSOA methodology implements the SFOA method to fine-tune hyperparameters, ensuring the selection of optimal values that improve performance. This adaptive tuning process enhances model accuracy and generalization. It also mitigates manual intervention and speeds up convergence during training, resulting in a more efficient learning process.
- This HDLID-ECSOA method introduces a novel methodology by integrating DOA-based feature selection, a CNN-BiGRU classifier enhanced with CrAM, and SFOA-based hyperparameter tuning. This integration enables more accurate, efficient, and context-aware classification, demonstrating significant novelty in model design.

Review of literature

Chen et al.[11] proposed a model to overcome over-parameterization in present optimization-based heuristic models, the geometrized task scheduling concern is handled by changing the distribution of clustered challenges into a regional partition concern in a 2-D graph and implementing a Tetris-like challenge offloading approach for edge-cloud co-operation. An online learning model is used to fine-tune the sliding window length based on the developing circumstances. Al-Quayed et al.[12] introduce a secure decision-making technique employing reinforcement learning (RL) with the integration of BC to improve data protection and trust. The presented

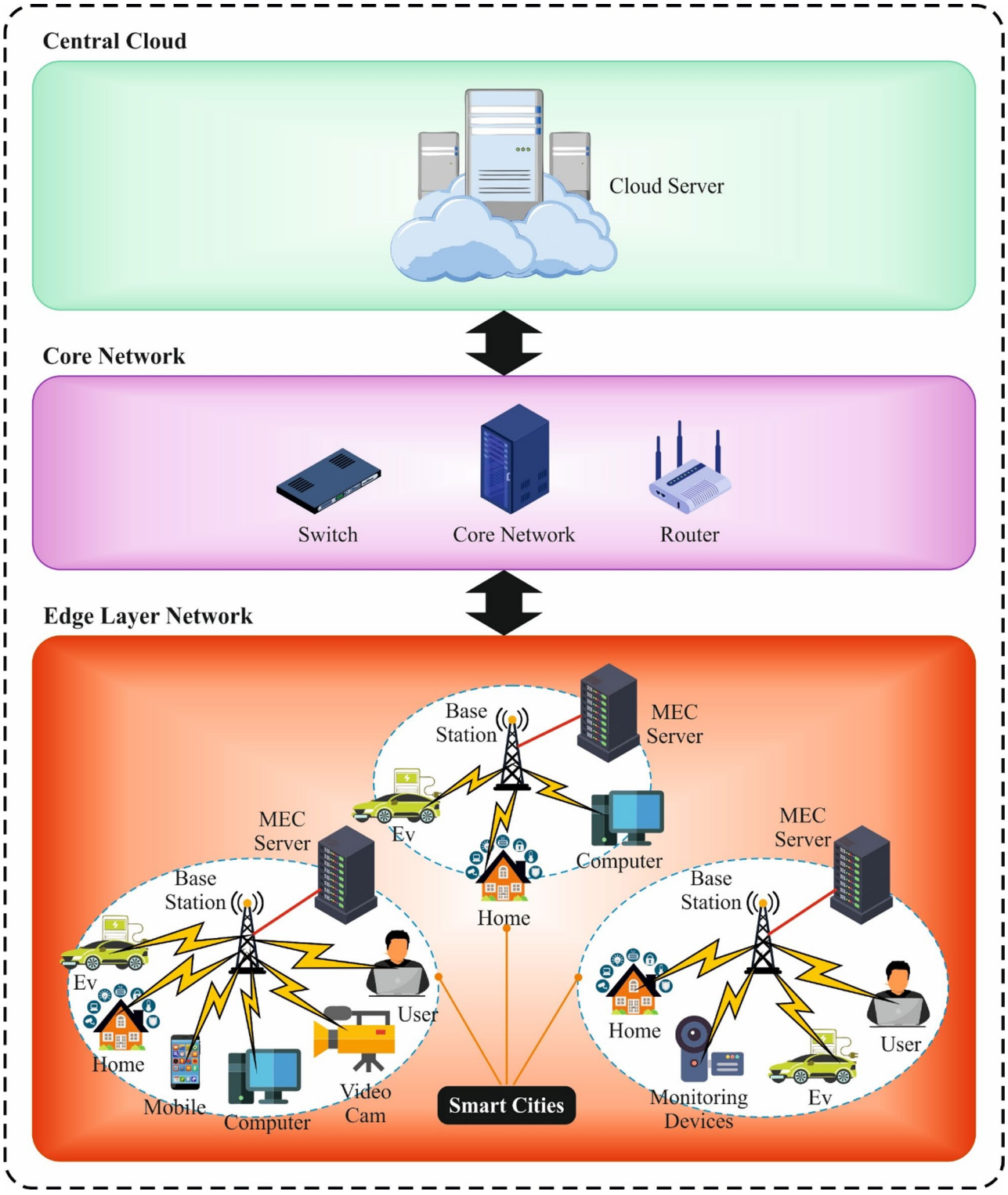

Fig. 1. Architecture of edge computing in smart cities.

approach raises the system efficacy for deploying sources and communication gadgets with security. It offers a dependable and more flexible model by investigating learning models to handle the instability and inaccuracy of cognitive methods. Sahu et al.[13] advanced a multi-objective optimizer framework for smart parking integrating digital twin (DT) technology, Markov decision process (MDP), particle swarm optimization (PSO), and pareto front optimizer (PFO). Thus, the projected structure employs DT. Additionally, PSO enhances the solution initiated from the Pareto front for a higher distribution. Chen et al.[14] developed an improved geospatial sensor web (GSW), integrating spatio-temporal modelling (STM) techniques and IoT protocols. Validated over the city

sensing base station (CSBS), a pilot experiment exhibited that the architecture incorporates different sensing sources through 8 protocols, accomplishing more than five stages with faster aerial-ground system formation in an emergency. In[15], a probability-based hybrid whale-dragonfly optimizer (p-H-WDFOA) edge-computing technique is proposed for smart urban vehicle transportation, decreasing edge-computing's wait and latency to tackle these problems. The 5G localized MEC servers. Tian et al.[16] developed a comprehensive security framework integrating an ELM-based replicator neural network (ELM-RNN) with the deep RL-based Deep Q-Networks (DRL-DQN) technique. Additionally, a secure trust-aware philosopher privacy and authentication (STAPPA) and garson algorithm (GA) is utilized for optimization to strengthen data protection and mitigate security breaches. Xu, Nagothu, and Chen[17] proposed an autonomous and resilient edge (AR-EC) framework that integrates SDN, blockchain (BC), and AI technologies. Software-defined networking (SDN) enables effective edge resource optimization and coordination with the assistance of AI methodologies like large language models (LLM). Moreover, a federated microchain fabric safeguards edge networks' resilience and security in a decentralized manner. Finally, a primary proof-of-concept prototype of an intelligent transportation system (ITS) data shows the possibility of implementing AR-Edge in real-time settings. In[18], a resource allocation method for hierarchical EC depending on attention mechanism (AM) is projected, to remove a smaller number of aspects that may depict services from a vast amount of data gathered from edge nodes. The AM is employed to define the precedence of service rapidly.

Wang et al.[19] developed NeuroSpatialIOT, an intuitive smart home control system that combines 2D spatial mapping, eye tracking, and DL to accurately interpret user intent. Far et al.[20] investigated the integration of BC and deep RL (DRL) models to optimize mobile transmission and secure data exchange in Iot-assisted smart cities, improving privacy, security, and system efficiency. Khan et al.[21] developed and evaluated energy-efficient parallel computation offloading mechanism through DL (EPCOD) and energy efficient DL-based offloading scheme (EEDOS) techniques, optimizing latency and energy usage in multi-task, multi-server mobile EC environments. Mishra and Chaurasiya[22] developed a hybrid DL LSTM-SVM approach integrating long short-term memory (LSTM) and support vector machine (SVM) classifiers. The system integrates min-max normalization for data preprocessing, feature selection using the reptile search algorithm (RSA), and BC technology to detect and prevent cyber-attacks effectively. Ficili et al.[23] explored and analyzed the integration of IoT, CC/EC, and AI to enable real-time decision-making and advance pervasive environmental intelligence. Wang et al.[24] proposed an AI-enhanced multi-stage learning-to-learning (MSLL) approach by utilizing MMStransformer for secure, efficient load management in IoT networks of smart cities, improving load forecasting accuracy while ensuring data security and privacy. Lilhore et al.[25] proposed a hybrid deep Q-network (DQN)-proximal policy optimization (PPO)-graph neural network (GNN)-RL model for optimizing resource allocation in dynamic IoT environments, enhancing efficiency, reducing costs, and improving performance in real-time applications. Ahmed and Elena[26] explored the integration of AI techniques such as ML, federated learning (FL), and intelligent orchestration frameworks with EC to enhance real-time decision-making, reduce latency, and improve scalability in smart cities, industrial IoT, and 5G/6G networks. Qasim Jebur Al-Zaidawi and Çevik[27] proposed a technique by utilizing hybrid grey wolf optimization with PSO (HGWOPSO) and hybrid world cup optimization with Harris Hawks optimization (HWCOAHHO) to symmetrically balance global exploration and local exploitation, optimizing DL models for real-time anomaly detection in IoT networks. Additionally, a multi-criteria decision-making (MCDM) framework integrating the analytic hierarchy process (AHP) and technique for order preference by similarity to ideal solution (TOPSIS) is employed to effectively evaluate and rank the proposed methods. Kumar and Neduncheliyan[28] developed an ensemble DL methodology integrating self-attention CNN, BiGRU, and shark smell optimized feed forward networks (SSOFFN) techniques for improved cybersecurity in IoT-based smart cities, utilizing fog computing (FC) to mitigate latency and computational overhead. Table 1 summarizes the existing studies on intelligent EC using DL and optimization techniques.

The existing studies on IoT-based smart city applications exhibit crucial improvements in security, resource optimization, and load management. However, several approaches concentrate on optimizing a single aspect, such as energy consumption or security, without considering a holistic, multi-objective optimization. The integration of AI, BC, and EC remains fragmented, with restricted exploration of their coordination. Furthermore, several techniques, such as DQN-PPO-GNN and EPCOD, show limitations in addressing the scalability problems when dealing with massive, dynamic IoT environments. Moreover, the adaptability of hybrid models like LSTM-SVM in real-time anomaly detection is often unexamined in diverse urban contexts. Another research gap is the lack of standardized frameworks for secure and effective data exchange in multi-server, multi-task settings across various IoT applications. Additionally, existing models often fail to integrate privacy-preserving mechanisms crucial for protecting sensitive data in decentralized IoT networks. The absence of unified evaluation benchmarks also affects objective comparison and validation of proposed approaches across diverse smart city scenarios.

The proposed method

This manuscript presents the HDLID-ECSOA model. This technique's main goal is to provide intelligent EC in smart cities using advanced optimization models. It contains data pre-processing, a DOA-based FS process, hybrid classification, and a parameter fine-tuning process. Figure 2 depicts the entire flow of the HDLID-ECSOA technique.

Data normalization: Min-Max

Initially, the data pre-processing employs the min-max normalization method to convert and standardize raw data to improve the efficiency of models[29]. This model is chosen due to its simplicity and efficiency in scaling data within [0, 1]. This technique ensures that all features contribute equally to the model, preventing any single feature from dominating due to differences in scale. This technique is less sensitive to outliers and performs well when the data is not normally distributed, compared to other normalization techniques, such as

Reference Number	Objective	Method	Dataset	Measures
Chen et al[11]	For efficient task scheduling and resource allocation for large-scale EC in smart cities.	Regional partitioning, voronoi diagrams, tetris offloading, adaptive allocation, online learning	Large-scale EC workloads	Task deadline violation rate
Ai-Quayed et al.[12]	To improve secure and efficient routing in cognitive networks for smart cities using RL and BC.	RL, BC, fault detection and validation, adaptive resource management	Simulated cognitive network data	Network efficiency, security metrics
Sahu et al.[13]	To optimize smart parking management using a multi-objective framework integrating DT and hybrid optimization models.	DT, MDP, PSO, PFO	Simulated smart parking data	Search time, energy usage, traffic congestion
Chen et al[14]	To enhance GSW for unified, collaborative edge-side observation and dynamic resource management.	IoT protocol integration, stm, sensorml and sos extension, dynamic task allocation	City sensing base station data	Resource accessibility, real-time processing
Wang et al.[15]	To reduce latency and delay in EC for smart urban vehicle transportation using hybrid optimization.	p-H-WDFOA, 5G multi-access EC, hybrid edge-cloud architecture	Smart urban vehicle network data	Latency, processing time, energy consumption
Tian et al.[16]	To enhance security and anomaly detection in IIoT EC for smart cities using hybrid learning models.	ELM-RNN, DRL-DQN, distributed authorization mechanism, STAPPA, GA	Simulated IIot network data	Detection rate, accuracy
Xu, Nagothu, and Chen[17]	To develop an autonomous and resilient EC architecture integrating AI, SDN, and BC for secure and efficient IoT ecosystems.	AR-EC, SDN, BC, LLM, federated microchain blockchain, autonomous edge resource coordination	ITS data	Network resilience, security, resource optimization
Sun et al.[18]	To optimize resource allocation and task scheduling in hierarchical EC networks for smart cities to ensure quality of service.	AM, priority determination, Q-Learning	Simulated edge network data	Resource utilization, task delay, QoS guarantee
Wang, Wang, and Du[19]	To develop an intuitive system to assist users with severe disabilities in smart home environments.	NeuroSpatialIOT, Eye Tracking, DL, context-aware control display	Eye movement and Spatial data from users	Usability score, task completion time
Far et al.[20]	To explore the integration of BC and DRL to enhance privacy, security, and transmission efficiency in IoT-assisted smart cities.	BC, DRL, IoT system clustering and categorization, taxonomy development	Literature review (2015–2024)	Privacy, security, transmission efficiency
Khan et al.[21]	To develop an energy-efficient DL-based mechanism for optimal parallel computation offloading in mobile EC.	EPCOD, EEDOS, multi-factor analysis	Large simulated offloading data	Latency, energy consumption, accuracy
Mishra and Chaurasiya[22]	To develop a hybrid DL model to secure IoT-based smart city transactions by detecting and preventing cyber-attacks.	LSTM-SVM, Min-Max normalization, weighted average filtering, RSA, BC	Smart city transaction data	Accuracy, specificity, f1 score
Ficili et al.[23]	To explore and analyze innovative integration approaches to enable real-time decision-making and predictive analytics.	IoT Integration, CC, EC, AI	Case studies and applications	Latency, decision accuracy
Wang et al.[24]	To develop an AI-based model for secure and accurate load management in IoT-based smart cities.	MSLL, MMStransformer, load forecasting, security integration	Smart city IoT data	Prediction accuracy, computational efficiency
Lilhore et al.[25]	To develop a hybrid model for dynamic and efficient resource allocation and workload scheduling in IoT edge-cloud ecosystems.	DQN, PPO, GNN, RL	Google cluster data, alibaba cluster trace, microsoft azure traces	Scheduling time, operational cost, energy consumption
Ahmed and Elena[26]	To investigate how integrating AI with EC enhances real-time decision-making, scalability, and network resilience in autonomous networks.	ML, FL, intelligent orchestration	Various real-world IoT and network data	Latency, scalability, resource utilization
Qasim Jebur Al-Zaidawi and Çevik[27]	To optimize DL for real-time IoT anomaly detection with balanced scalability and efficiency using hybrid optimization.	HGWOPSO, HWCOAHHO, MCDM, AHP	Real-world IoT network data	Accuracy, precision, recall, f1-score
Kumar and Neduncheliyan[28]	To develop an ensemble DL model for accurate IoT cyber-attack detection.	CNN, BiGRU, SSOFFN, FC	ToN-IoT dataset	Detection rate, auc, accuracy, precision, recall, f1-score

Table 1. Performance comparison of the proposed ensemble DL model with existing methods for IoT cyber-attack detection in smart City environments.

Z-score normalization. This benefits datasets with varied feature ranges, like those found in IoT applications. Furthermore, this normalization method is computationally efficient, making it ideal for massive datasets, ensuring faster model convergence during training.

Min-max normalization is carried out on the data to scale the vectors in standardization. Equation (1) provides the design of min-max normalization.

$$y' = \frac{(V - \min)}{(\max - \min)} \tag{1}$$

Here, y denotes a set of features, $\max$ signifies the maximal value from the features, and min denotes the minimal value in the features of the dataset V'. This characterizes the standardized data, which holds values from $(0–1)$.

Feature selection: DOA

Furthermore, the DOA implements a subset of the FS process to detect and choose the most relevant features from the input data[30]. This model is selected for its capability to effectively explore the search space and detect

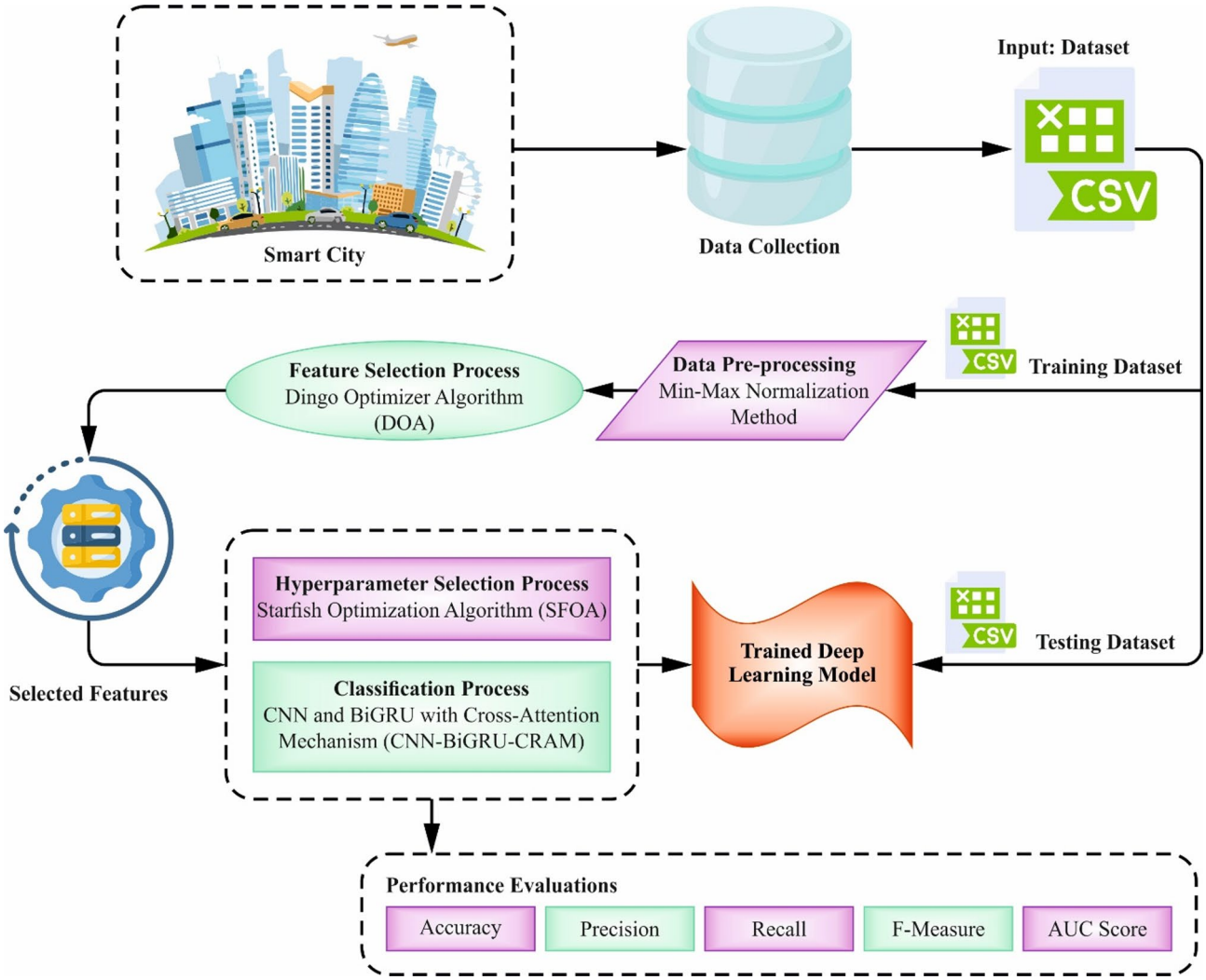

Fig. 2. Overall flow of HDLID-ECSOA model.

the most relevant features. This population-based optimization technique replicates the natural behaviour of dingo packs, which assists in avoiding local minima and finding optimal feature subsets. Unlike conventional methods such as genetic algorithms or PSO, DOA is prevalent because of its faster convergence and capability to handle high-dimensional data. Its flexibility in balancing exploration and exploitation makes it appropriate for feature selection tasks in complex datasets. Moreover, DOA does not require gradient data, making it suitable for nonlinear and non-convex feature selection problems. This approach improves the model's performance by eliminating redundant or irrelevant features, enhancing accuracy and reducing computational cost. Figure 3 illustrates the workflow of the DOA technique.

DOA simulates the social behaviour of Australian dingos. This model is stimulated by dingoes' searching tactics, which are attacked by scavenging behaviour, grouping tactics, and persecution. Three search approaches related to the four rules are established to increase the model's efficacy and performance.

The initial approach is a group attack. Predators frequently utilize very intellectual searching methods. Dingoes generally search for smaller prey like rabbits separately; however, they collect in groups after hunting larger prey like kangaroos. It may discover the prey position and encircle it, like wolves; this behaviour is characterized by the succeeding Eq. (2):

$$\overrightarrow{x}_i(t+1) = \beta_1 \sum_{k=1}^{n\alpha} \frac{\left[\overrightarrow{\varphi_k(t)} - \overrightarrow{x_i}(t)\right]}{na} - \overrightarrow{x_*}(t) \tag{2}$$

Whereas $\overrightarrow{x}_i(t+1)$ refers to the new location of the searching agent, na denotes a randomly generated integral number in the range [2, SizePop/2]. In contrast, SizePop stands for total dimensions (dingoes that assault). At the same time, $\varphi_k(t)$ denotes sub-set of searching agent, X refers to dingoes' population randomly made, $x(t)$ signifies present search iteration, and $\beta 1$ signifies uniformly generated random number from interval $[-2,2]$, $x(t)$ symbolize top searching agent identified from the preceding iteration. The second tactic is Persecution.

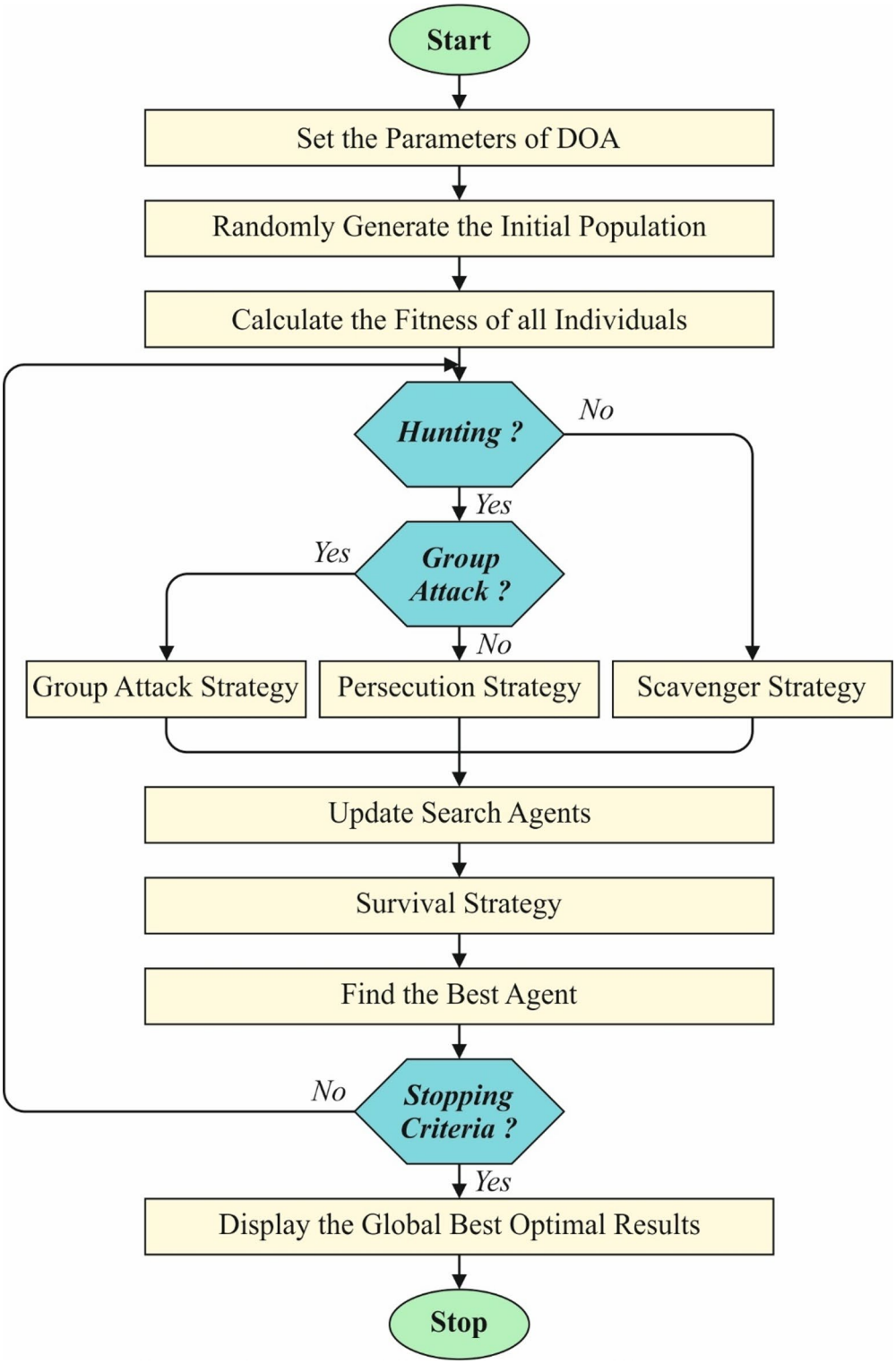

Fig. 3. Working flow of the DOA methodology.

Dingos generally search for smaller prey that is hunted till it is captured separately. The following Eq. (3) models this behaviour:

$$\overrightarrow{x}_i(t+1) = \overrightarrow{x}_{*(t)+\beta_1 * e^{\beta_2 * (}} \overrightarrow{x}_{r_1}(t) - \overrightarrow{x}_i(t)) \tag{3}$$

Whereas β_2 denotes uniformly generated random numbers within the range $[-1, 1]$, r_1 refers to randomly generated numbers from 1 to the dimensions of a maximum of search agents (dingoes), and $i \neq r_1$.

The third tactic is Scavenger, whose behaviour is described as the action once dingos discover corpses to feed on after arbitrarily walking in their environment. The succeeding Eq. (4) models this behaviour:

$$\overrightarrow{x}_i(t+1) = \frac{1}{2}[e^{\beta_2} * \overrightarrow{x}_{r_1}(t) - (-1)^{\sigma} * \overrightarrow{x}_i(t)] \tag{4}$$

Whereas σ denotes a binary number randomly ϵ {0,1}, and $i \neq r_1$.

Fourth tactic: Survival Rates of Dingos: Dingo dogs are at risk of extinction because of illegal hunting. The dingoes' survival rate value is provided by the succeeding Eq. (5):

$$\overrightarrow{x}_i(t) = \overrightarrow{x}_*(t) + \frac{1}{2}[\overrightarrow{x}_{r_1}(t) - (-1)^{\sigma} * \overrightarrow{x}_{r_2}(t)] \tag{5}$$

The fitness function (FF) reflects the accuracy of classification and the number of selected features. It utilizes classification accuracy and reduces the preferred features' dimensionality. Hence, the FF below is deployed to assess discrete solutions.

$$Fitness = \alpha * ErrorRate + (1-\alpha) * \frac{\#SF}{\#All\_F} \tag{6}$$

Here, $ErrorRate$ is the classifier rate of error utilizing the chosen features. $ErrorRate$ is calculated as the proportion of improperly classified to several classifications prepared among 0 and 1. $\#SF$ means several preferred features, and $\#All\_F$ refers to the complete number of features in the original dataset. α is employed to control classifier excellence and subset length. The value of α is 0.9.

Classification process: CNN-BiGRU-CrAM

Besides, the proposed HDLID-ECSOA model employs the CNN-BiGRU-CrAM technique for the classification process[31]. This model was chosen for its capability of effectively capturing both spatial and sequential dependencies in data. The CNN shows excellence in extracting hierarchical features from raw input, particularly for tasks comprising spatial patterns, such as image or time-series data. The BiGRU layer is appropriate for processing sequential data, as it captures context from past and future time steps, improving temporal feature representation. Integrating the CrAM enhances the model's focus on the most relevant features by allowing it to learn contextual relationships between diverse input parts. This integration of CNN, BiGRU, and CrAM outperforms conventional methods by giving a more holistic representation of data. It also improves classification accuracy by addressing spatial and temporal dimensions in intrinsic datasets, making it ideal for IoT and IIoT applications. Figure 4 depicts the infrastructure of CNN-BiGRU-CrAM method.

CNN is a deep-structured feedforward NN that contains a convolution calculation. The sparsity of links among layers and sharing the hidden layer (HL)'s convolutional kernel framework permit CNN to remove features with minimum calculation. The fully connected (FC), convolutional, input, and pooling layers are presented in the complete architecture of CNN. During this input layer, the removed attributes are provided as input. In contrast, the convolutional layer utilizes convolution calculations on the input with the convolutional kernel to obtain feature mapping. The integration involving Z convolutional kernels is indicated by $[Q_1, Q_2, Q_3, \ldots, Q_Z]$, whereas Q_Z imitates the Zth kernel size of the convolution or the longitudinal dimension of the convolution kernel window. There would be Z feature mapping vectors afterwards, computing Z convolutional kernels. The size of the convolution window was provided as I. To remove the local attributes, the convolutional kernel is applied to perform the convolution process on the input windows y_1^H, y_2^H, y_3^H, y_{p-H+1}^p. Assuming that the input D contains p feature vectors, y_1, y_2, y_3, y_p, that is represented as

$$z_j = g(X.y_{j:j+H-1} + B) \tag{7}$$

.

Whereas $g()$ indicates the nonlinear function, H suggests the kernel size of the convolution, and B and X indicate the biased vector and weighted matrix. The vector's incorporation is signified by $y_{j:j+H-1}$.

After the extraction of the convolutional kernel, the eigen-vector z is gained as:

$$z = \{z_1, z_2, z_3, \ldots, z_{p-H+1}\} \tag{8}$$

Following the convolution process, every eigen-vector is vulnerable to pooling processing using the pooling layer. The multi-dimensional vectors are converted into a value afterwards; pooling is applied as the pooled vector component. The larger component in the series z_1, z_2, z_3, z_{p-H+1} should be selected by the maximal pooling model, which would lastly give novel vectors z:

$$z = \max(z_j) \tag{9}$$

A kind of RNN, such as LSTM, is considered GRU. RNNs implement recursion in the progressive direction of the sequence by inputting sequential data. Therefore, RNN contains memory and parameter sharing. Moreover, RNNs perform well while learning nonlinear features from sequential data. LSTM can learn data of correlation between long-term and short-term sequential data. Subsequently, GRU was recognized as the solution to the LSTM problem, using a slow convergence rate and extreme parameters. The inner modules of GRU contain the reset and update gates.

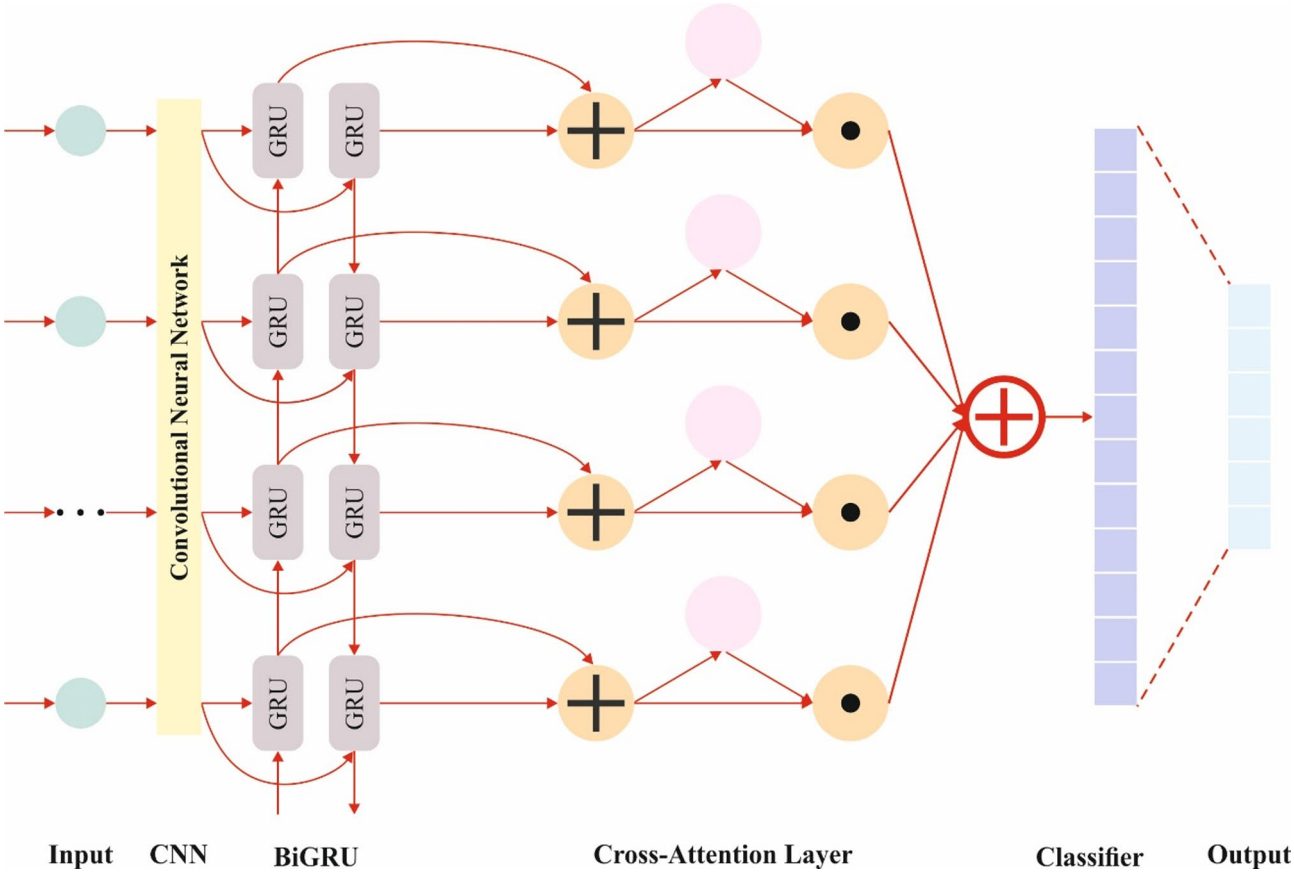

Fig. 4. Structure of CNN-BiGRU-CrAM model.

This framework allows the Bi-GRU to successfully handle the flow of information by establishing which feature to retain or discard. Compared with LSTM, GRU exchanges the input and forget gates of the LSTM using an update gate. The value of the greater update gate imitates the greater influence degree. The succeeding equation is applied to describe the HL component B:

$$Z_u = \delta\ (w_Z.[H_{u-1},\ y_u]) \tag{10}$$

$$R_u = \delta\ (w_R.[H_{u-1},\ y_u]) \tag{11}$$

$$\widetilde{H}_u = \tanh\left(w\left[R_u * H_{u-1},\ y_u\right]\right) \tag{12}$$

$$H_u = (1 - Z_u) * H_{u-1} + Z_u * \widetilde{H}_u \tag{13}$$

Whereas δ specifies the function of Sigmoid, $\tanh$ indicates the function of hyperbolic tangent, and Z_u and R_u identify the reset and update gates. w_R, v_R, wz, vz, and v imply each of the training parameter matrices. The reset R_u, the input gate Y_u, in the present sample, the output H_{u-1} of the HL neuron at the previous instant, and the matrices of training parameters v and u work together to determine the activation state of the candidate $\widetilde{H}_u$ at the current instant. The Bi-GRU network is superior for learning the connections among present, previous, and upcoming determinant factors. It is measured as

$$z_2 = g\left(vB_2 + v'\,B_2'\right) \tag{14}$$

.

and B_2' are calculated utilizing

$$B_2 = g\left(wB_1 + uy_2\right) \tag{15}$$

$$B_2' = g\left(w'\,B_3' + uy_2\right) \tag{16}$$

.

The HL value S_u in the forward computation is related to S_{u-1}. The HL value S_u in the backwards calculation is associated with S_{u-1}. The forward and reverse calculations determine the last outcome. The computation process of the bi-directional RNN is provided as shown:

$$O_u = g\left(vS_u + v'\, S'_u\right) \tag{17}$$

$$S_u = f\left(uy_u + wS_{u-1}\right) \tag{18}$$

$$S'_u = f\left(u'\, y_u + w'\, S'_{u-1}\right) \tag{19}$$

Cross-entropy (CE) loss function is applied for the classification difficulty within the NN.

The AM is designed to reduce the intervention of incorrect data and assist make the most promising results. The AM works well in dual methods: Initially, it spontaneously recognizes the local data which needs to be focused on the global input. Due to these dual advantages, AM is often applied to enhance local data features. Moreover, the attention region and the task objective are dissimilar. The significant data is used effectively after local information is collected. The attention network output is specified as shown:

$$Z_u = g\left(z_{u-1},\ m_{u-1},\ D_u\right) \tag{20}$$

Here, Z_u symbolizes the AM's output at a sample u, g indicates the dense layer, Z_{u-1} suggests the AM's output at time $u-1$, and m_{u-1} defines the label at instant $u-1$.

$$D_u = \sum_{k=1}^{U_y} b_{uk} i_k \tag{21}$$

.

D_u symbolizes the following stage output, i_k involves the AM's kth input, and b_{uk} describes the attention weights.

$$b_{uk} = \frac{\exp\left(e_{uk}\right)}{\sum_{k=1}^{U_y} \exp\left(e_{ul}\right)} \tag{22}$$

$$e_{uk} = h\left(z_{u-1},\ i_k\right) = w \times \tanh\left(X \times\ i_k + v \times\ Z_{u-1} + c\right) \tag{23}$$

Meanwhile, h is applied to calculate the area of relation between Z_{u-1} and i_k, and b_{uk} defines the amount to which the present AM is associated with the kth input.

Parameter Fine-Tuning: SFOA

The SFOA model is utilized for hyperparameter tuning to optimize model performance to ensure that the optimum hyperparameters are selected for enhanced accuracy[32]. This model is chosen due to its robust global search capability and efficiency in finding optimal solutions. The model is inspired by the foraging behaviour of starfish, which makes it perform well in balancing exploration and exploitation effectively, making it less likely to get trapped in local minima than conventional optimization techniques like gradient descent. Unlike grid or random search methods, SFOA can adaptively adjust its search process to find the optimum hyperparameters with fewer iterations. This results in an enhanced optimization accuracy and mitigated computational cost. Additionally, this model is appropriate for high-dimensional, complex parameter spaces commonly found in DL models. Its flexibility and robustness across diverse tasks make it an ideal choice for fine-tuning parameters, ensuring the model attains optimal performance without excessive computational overhead. Figure 5 represents the framework of the SFOA approach.

Meta-heuristic methods frequently balance exploitation (local search refinement) and exploration (global search capability). Efficient optimization must be the best tradeoff among these dual stages to avoid early convergence to sub-optimal solutions. As meta-heuristic tactics depend on randomized searching mechanisms, they do not guarantee the best solutions for all problems. The bio-inspired design for SFOA is acquired from starfish's hunting, movement, and regenerating capabilities. Starfish, otherwise named sea stars, usually show a five-arm radial symmetry extending from the primary disk.

The exploration stage of SFOA imitates the starfish's foraging behaviour, whereas exploitation is modelled through regeneration and predation tactics. SFOA uses a hybrid searching mechanism that combines:

- For 5-dimensional search $(L > 5)$, stimulated by the five arms of the starfish, for different exploration.
- Increasing convergence if the feature area is small for a 1-dimensional search (L ≤ 5).

The optimizer procedure of SFOA contains three main phases:

Initialization: At the start of the optimizer procedure, starfish locations are randomly made inside the pre-defined design area, expressed as:

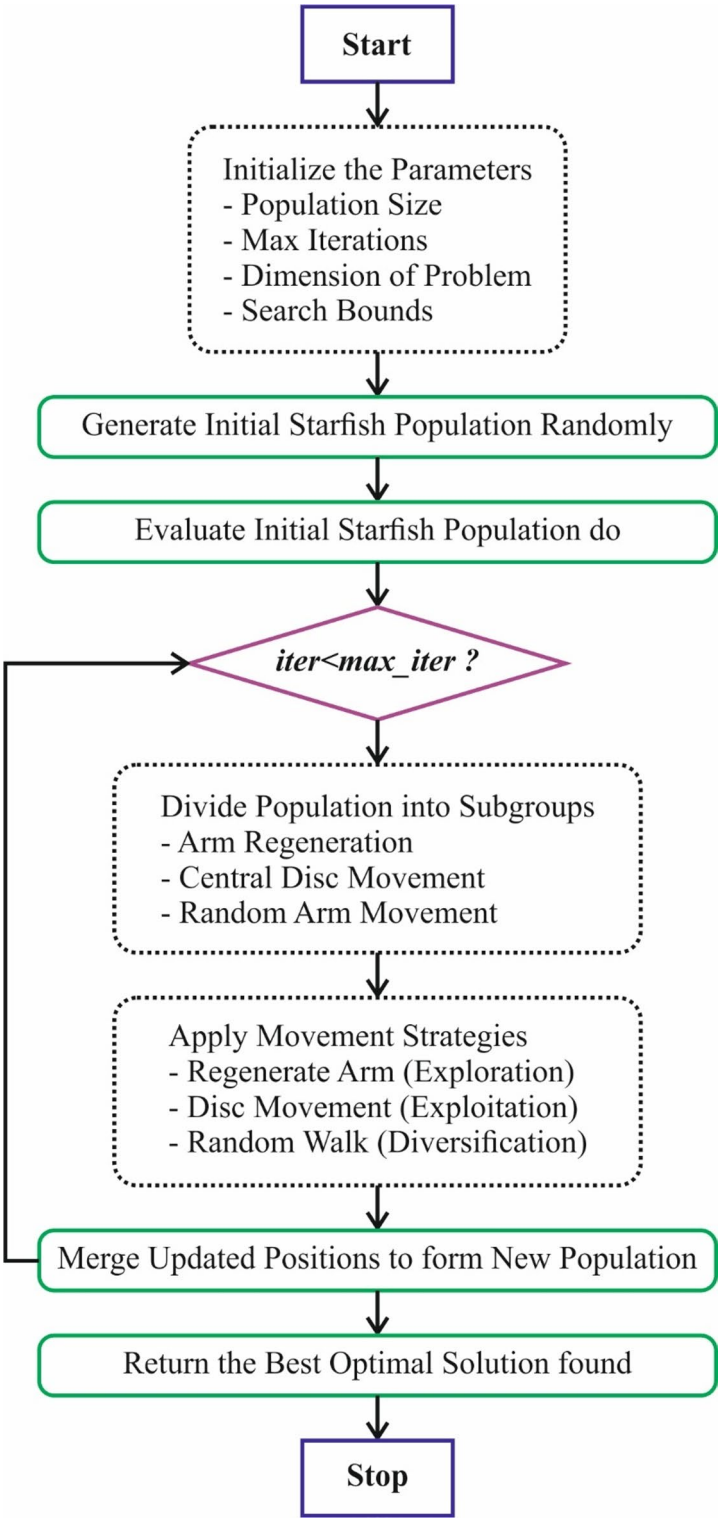

Fig. 5. SFOA framework.

$$X = \begin{bmatrix} \chi_{11} & x_{12} & \cdots & x_{1L} \\ x_{21} & x_{22} & \cdots & x_{2L} \\ \chi_{31} & x_{32} & \cdots & x_{3L} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NL} \end{bmatrix} \tag{24}$$

Whereas N denotes the size of the population, L refers to the design variable counts, and the primary locations are calculated as:

$$x_{ij} = low_j + random\left(upper_j - low_j\right),\ i = 1,2,\ \ldots\ , N,\ j = 1,2,\ \ldots\ ,\ L \tag{25}$$

The fitness score of all starfish is estimated according to the objective function, allowing an adaptive searching process.

Exploration: SFOA uses dissimilar tactics according to the dimensionality of the problem:

- For $L > 5$, a 5-dimensional search is applied for larger-scale optimization.
- For $L \leq 5$, a 1-dimensional search was utilized for enhanced local refinement.

The location updated rule in the exploration stage is expressed as:

$$X_{i,p}^{t+1} = \begin{cases} X_{i,p}^{t} + a\left(X_{best,p}^{t} - X_{i,p}^{t}\right)\cos\theta\ ,\ if\ rn \leq\ 0.5, \\ X_{i,p}^{t} + a\left(X_{besi,p}^{t} - X_{i,p}^{t}\right)\sin\theta\ ,\ if\ rn > 0.5. \end{cases} \tag{26}$$

Whereas rn denotes randomly generated numbers $e\,(0,1)$, and $X_{i,p}^{t+1}, X_{i,p}^{t}$, and $X_{besi,p}^{t}$ characterize the computed, present, and best locations, correspondingly. The parameters a and θ are provided by:

$$a = (2r-1)\,\pi\ ,\ \theta\ = \frac{\pi}{2} \cdot \frac{t}{t_{\max}} \tag{27}$$

When the modified position is outside the borders of the model parameters, the arms are given to stay in the preceding location instead of migrating. The exploration stage upgrades the area utilizing the unidimensional search design if $L \leq\ 5$. In such a case, a starfish uses location data from others to move one of its arms toward the food resource:

$$X_{i,p}^{t+1} = E_t X_{i,p}^{t} + a_1\left(X_{k1,p}^{t} - X_{i,p}^{t}\right) + a_2\left(X_{k2,p}^{t} - X_{i,p}^{t}\right) \tag{28}$$

Here, $X_{k1,p}^{t}$ and $X_{k2,p}^{t}$ are randomly chosen P-dimensional positions from dual starfish, a_1 and a_2 are randomly generated numbers $e\,(-1,1)$, and E_t is measured as:

$$E_t = \frac{t_{\max} - t}{t_{\max}} \cos\theta \tag{29}$$

Exploitation: The exploitation stage consists of regeneration and hunting tactics. The location of the starfish is upgraded according to the best position:

$$d_n = X_{best}^{t} - X_{n_p}^{t},\ n = 1,2,\ \ldots\ ,5 \tag{30}$$

Edge-IIoT Dataset	
Class Labels	**No. of Record**
Normal	3000
DDoS-UDP	3000
DDoS-ICMP	3000
SQL injection	3000
DDoS-TCP	3000
Password	3000
DDoS-HTTP	3000
Uploading	3000
Backdoor	3000
XSS	3000
Ransomware	3000
Fingerpriniting	3000
Total Record	**36,000**

Table 2. Details of Edge-IIoT dataset.

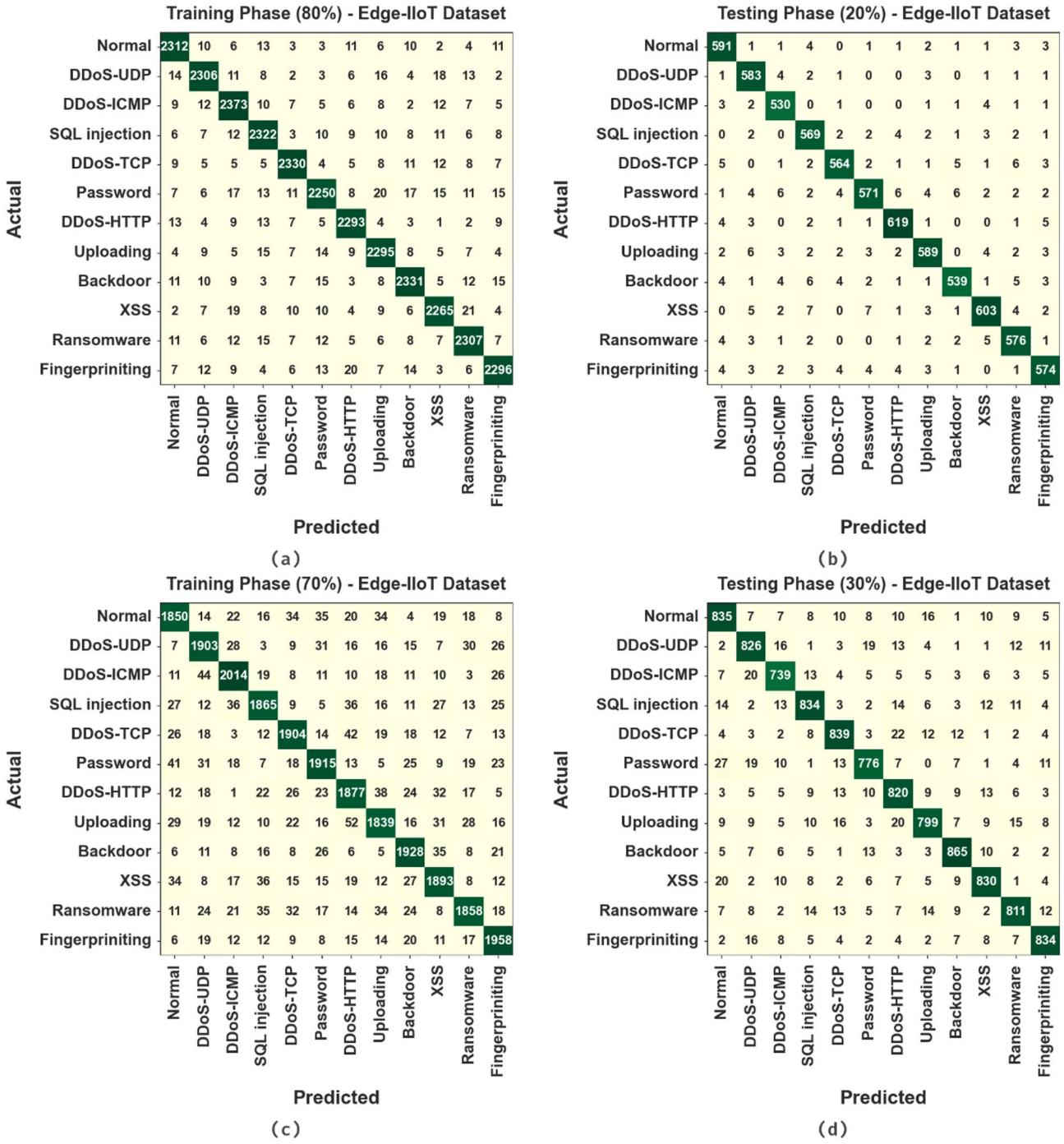

Fig. 6. Edge-IIoT dataset (a-b) 80%TRAPHA and 20%TESPHA and (c-d) 70%TRAPHA and 30%TESPHA.

The new location is calculated utilizing:

$$X_i^{t+1} = X_i^t + rn_1 d_{n1} + rn_2 d_{n2} \tag{31}$$

Here, rn_1 and rn_2 signify randomly generated values in $(0,1)$, and d_{n1}, d_{n2} denote randomly selected distances.

Moreover, in the regeneration stage, if a starfish loses an arm to prevent predators, the position is upgraded utilizing:

$$X_i^{t+1} = \exp\left(-t \times \frac{N}{t_{\max}}\right) X_i^t \tag{32}$$

The last upgrade guarantees that values remain inside limits:

$$X_i^{t+1} = \begin{cases} X_i^{t+1}, & if\ low_i \le X_i^{t+1} \le upper_j, \\ low_j, & if\ X_i^{t+1} < low_j, \\ upper, & if\ X_i^{t+1} > upper_i. \end{cases} \tag{33}$$

The SFOA generates a feature set (FF) to improve the classification accuracy. It defines an optimistic number to signify the better productivity of the candidate solution. The classifier rate of error minimization was measured as FF, as set in Eq. (34).

$$fitness\,(x_i) = ClassifierErrorRate\,(x_i) = \frac{no.\ of\ misclassified\ instances}{Total\ no.\ of\ instances} \times 100 \tag{34}$$

Experimental validation

The simulation validation of the HDLID-ECSOA model is examined under the Edge-IIoT dataset[33]. This dataset holds 36,000 records under 12 types of events, as depicted in Table 2. There are 63 features, but only 47 are selected.

Figure 6 presents the confusion matrix created by the HDLID-ECSOA technique using the Edge-IIoT dataset under an 80:20 training phase (TRAPHA), testing phase (TESPHA), and 70:30 TRAPHA/TESPHA. The outcomes specify that the HDLID-ECSOA method efficiently identifies and detects all classes precisely.

Table 3; Fig. 7 illustrate the intrusion detection of the HDLID-ECSOA methodology under the Edge-IIoT dataset. Depending on 80% TRAPHA, the proposed HDLID-ECSOA approach attains an average $accu_y$ of 99.35%, $prec_n$ of 96.11%, $reca_l$ of 96.11%, $F_{Measure}$ of 96.11%, AUC_{Score} of 97.88%, and Kappa of 97.95%. Besides, based on 20% TSAPHA, the proposed HDLID-ECSOA approach achieves an average $accu_y$ of 99.32%, $prec_n$ of 95.95%, $reca_l$ of 95.95%, $F_{Measure}$ of 95.94%, AUC_{Score} of 97.79%, and Kappa of 97.86%. Likewise, based on 70% TRAPHA, the proposed HDLID-ECSOA approach attains an average $accu_y$ of 98.42%, $prec_n$ of 90.49%, $reca_l$ of 90.49%, $F_{Measure}$ of 90.48%, AUC_{Score} of 94.81%, and Kappa of 94.88%. Also, based on 30% TSAPHA, the proposed HDLID-ECSOA approach attains an average $accu_y$ of 98.47%, $prec_n$ of 90.82%, $reca_l$ of 90.81%, $F_{Measure}$ of 90.80%, AUC_{Score} of 94.99%, and Kappa of 95.06%.

In Fig. 8, the training (TRAN) $accu_y$ and validation (VALN) $accu_y$ outcomes of the HDLID-ECSOA methodology under the Edge-IIoT dataset below 80:20 are demonstrated. The figure highlights that both values of $accu_y$ demonstrate rising trends that informed the capability of the HDLID-ECSOA methodology with maximum performance over several iteration counts. Besides, both $accu_y$ leftovers closer over the epochs, which indicates minimal overfitting and exhibits superiority of the HDLID-ECSOA model, guaranteeing consistent prediction on unseen instances.

In Fig. 9, the TRAN loss (TRANLOS) and VALN loss (VALNLOS) graph of the HDLID-ECSOA model under the Edge-IIoT dataset below 80:20 is exhibited. It is indicated that both values illustrate a falling trend, notifying the capacity of the HDLID-ECSOA model to balance a tradeoff between generalization and data fitting. The constant reduction in loss values further ensures the improved effectiveness of the HDLID-ECSOA model and refines the prediction outcomes.

Table 4; Fig. 10 show the comparative results of the HDLID-ECSOA methodology under the Edge-IIoT dataset with existing methods[19,35–38]. The outcomes emphasized that the LSTM, Random Forest (RF), FL, FedMLDL-Bayesian HPO, LDA, Gradient Boosting (GB), and J48 methodologies obtained lesser performance. The CNN-LSTM, DBN, and NeuroSpatialIOT methodologies also attained slightly lesser results. However, the proposed HDLID-ECSOA method reported superior performance with higher $accu_y$, $prec_n$, $reca_l$ and $F_{Measure}$ of 99.35%, 96.11%, 96.11%, and 96.11%, correspondingly.

Table 5; Fig. 11 specifies the computational time (CT) analysis of the HDLID-ECSOA approach with existing models under Edge-IIoT dataset. The HDLID-ECSOA approach demonstrates the fastest performance with a CT of 7.63 s, outperforming all other approaches. LSTM records 10.93 s, RF at 11.12 s, and LDA model at 11.03 s, indicating relatively efficient CT with nearly equal CT. In contrast, FL takes 15.61 s, and CNN-LSTM attains 15.74 s, suggesting moderate computational efficiency. Among the slower models, DBN shows 16.79 s while FedMLDL-Bayesian HPO illustrates the highest CT of 17.20 s. NeuroSpatialIOT, J48, and GB record 12.70, 13.20, and 12.39 s respectively.

Table 6; Fig. 12 portrays the error analysis of the HDLID-ECSOA technique with existing methods under Edge-IIoT dataset. The GB model achieves the highest $accu_y$ of 10.96% with a $reca_l$ of 10.98% but a relatively low $prec_n$ of 5.45%, indicating it identifies relevant cases but struggles with $prec_n$. The RF method exhibits an $accu_y$ of 6.62% and the highest $F_{Measure}$ of 7.91%, suggesting balanced but modest performance. The

Classes	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$	AUC_{Score}	Kappa
TRAPHA (80%)						
Normal	99.40	96.13	96.70	96.41	98.17	98.24
DDoS-UDP	99.36	96.32	95.96	96.14	97.82	97.88
DDoS-ICMP	99.32	95.42	96.62	96.01	98.09	98.16
SQL injection	99.32	95.59	96.27	95.93	97.93	98.01
DDoS-TCP	99.48	97.08	96.72	96.90	98.23	98.30
Password	99.19	95.99	94.14	95.06	96.89	96.97
DDoS-HTTP	99.46	96.39	97.04	96.71	98.36	98.42
Uploading	99.34	95.74	96.35	96.05	97.98	98.06
Backdoor	99.34	96.24	95.97	96.10	97.81	97.89
XSS	99.34	96.14	95.77	95.95	97.71	97.77
Ransomware	99.33	95.97	96.00	95.99	97.82	97.88
Fingerpriniting	99.35	96.35	95.79	96.07	97.73	97.81
Average	99.35	96.11	96.11	96.11	97.88	97.95
TESPHA (20%)						
Normal	99.36	95.48	97.04	96.25	98.31	98.37
DDoS-UDP	99.39	95.11	97.65	96.36	98.60	98.66
DDoS-ICMP	99.47	95.67	97.43	96.54	98.53	98.61
SQL injection	99.29	94.68	96.77	95.71	98.14	98.19
DDoS-TCP	99.36	96.74	95.43	96.08	97.57	97.63
Password	99.15	96.29	93.61	94.93	96.64	96.70
DDoS-HTTP	99.46	96.72	97.17	96.95	98.43	98.50
Uploading	99.28	96.24	95.31	95.77	97.48	97.54
Backdoor	99.31	96.77	94.40	95.57	97.06	97.13
XSS	99.25	96.48	94.96	95.71	97.31	97.36
Ransomware	99.32	95.36	96.48	95.92	98.03	98.10
Fingerpriniting	99.25	95.83	95.19	95.51	97.41	97.48
Average	99.32	95.95	95.95	95.94	97.79	97.86
TRAPHA (70%)						
Normal	98.28	89.81	89.20	89.50	94.15	94.21
DDoS-UDP	98.39	89.72	91.01	90.36	95.03	95.09
DDoS-ICMP	98.62	91.88	92.17	92.03	95.70	95.75
SQL injection	98.39	90.84	89.58	90.21	94.38	94.44
DDoS-TCP	98.52	90.93	91.19	91.06	95.18	95.25
Password	98.37	90.50	90.16	90.33	94.64	94.71
DDoS-HTTP	98.17	88.54	89.59	89.06	94.27	94.34
Uploading	98.17	89.71	87.99	88.84	93.54	93.60
Backdoor	98.63	90.81	92.78	91.79	95.97	96.04
XSS	98.40	90.40	90.31	90.36	94.72	94.78
Ransomware	98.39	91.71	88.65	90.15	93.96	94.04
Fingerpriniting	98.67	91.03	93.19	92.10	96.18	96.25
Average	98.42	90.49	90.49	90.48	94.81	94.88
TESPHA (30%)						
Normal	98.23	89.30	90.17	89.74	94.58	94.66
DDoS-UDP	98.32	89.39	90.87	90.13	94.94	95.01
DDoS-ICMP	98.52	89.79	90.67	90.23	94.92	94.98
SQL injection	98.46	91.05	90.85	90.95	95.01	95.07
DDoS-TCP	98.56	91.10	92.00	91.54	95.58	95.66
Password	98.37	91.08	88.58	89.81	93.91	93.98
DDoS-HTTP	98.18	87.98	90.61	89.28	94.74	94.81
Uploading	98.27	91.31	87.80	89.52	93.52	93.59
Continued						

Classes	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$	AUC_{Score}	Kappa
Backdoor	98.84	92.71	93.82	93.26	96.56	96.64
XSS	98.64	91.92	91.81	91.86	95.54	95.61
Ransomware	98.47	91.85	89.71	90.77	94.49	94.57
Fingerpriniting	98.76	92.36	92.77	92.56	96.04	96.11
Average	98.47	90.82	90.81	90.80	94.99	95.06

Table 3. Intrusion detection of HDLID-ECSOA approach under Edge-IIoT dataset.

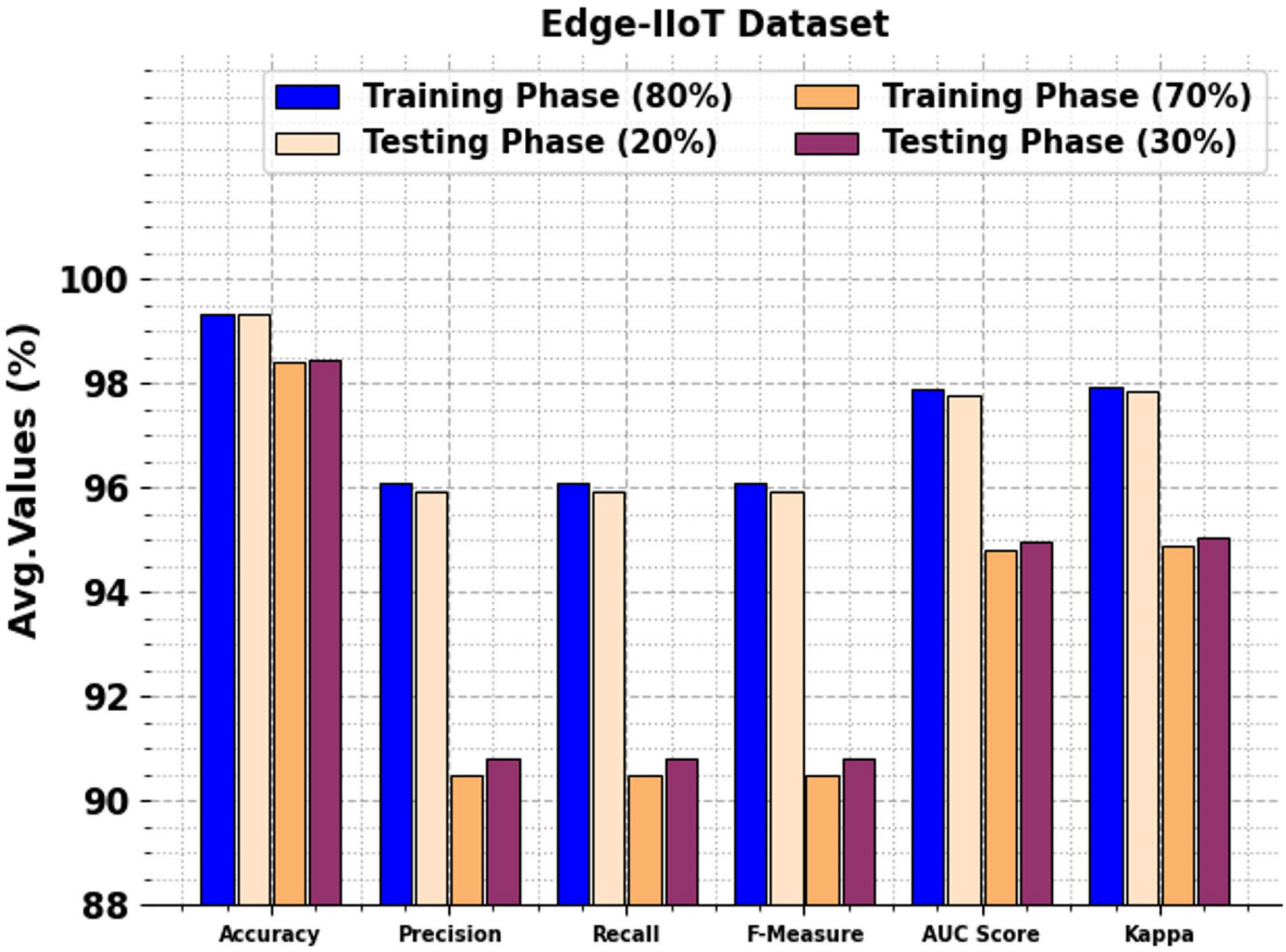

Fig. 7. Average of HDLID-ECSOA method under Edge-IIoT dataset.

LSTM model depicts an $accu_y$ of 5.17% with a $prec_n$ of 10.97% and $reca_l$ of 10.07%, highlighting robust detection capability despite low overall correctness. FedMLDL-Bayesian HPO and NeuroSpatialIOT illustrate $reca_l$ values of 10.98% and 10.45% respectively, showing potential in capturing relevant positives. However, the HDLID-ECSOA model significantly underperforms with only an $accu_y$ of 0.65% and equal $prec_n$, $reca_l$, and $F_{Measure}$ values of 3.89%, indicating limited utility in classification tasks within this context. Overall, while some models excel in specific metrics, none achieve uniformly robust performance across all indicators, highlighting challenges in IIoT error detection.

Table 7; Fig. 13 demonstrate the ablation study of the HDLID-ECSOA technique under the Edge-IIoT dataset. The baseline DOA approach attains an $accu_y$ of 97.23%, a $prec_n$ of 94.27%, a $reca_l$ of 94.01%, and an $F_{Measure}$ of 94.14%. Enhancing the optimization strategy with SFOA attains an $accu_y$ of 97.75%, a $prec_n$ of 94.79%, a $reca_l$ of 94.73%, and an $F_{Measure}$ of 94.69%. Incorporating DL with the CNN-BiGRU-CRAM model attains an $accu_y$ of 98.55%, a $prec_n$ of 95.57%, a $reca_l$ of 95.41%, and an $F_{Measure}$ of 95.33%. Finally, the proposed HDLID-ECSOA model attains the highest performance, with an $accu_y$ of 99.35%, a $prec_n$ of 96.11%, a $reca_l$ of 96.11%, and an $F_{Measure}$ of 96.11%, illustrating the strength of the integrated architecture in improving detection efficiency in IIoT environments.

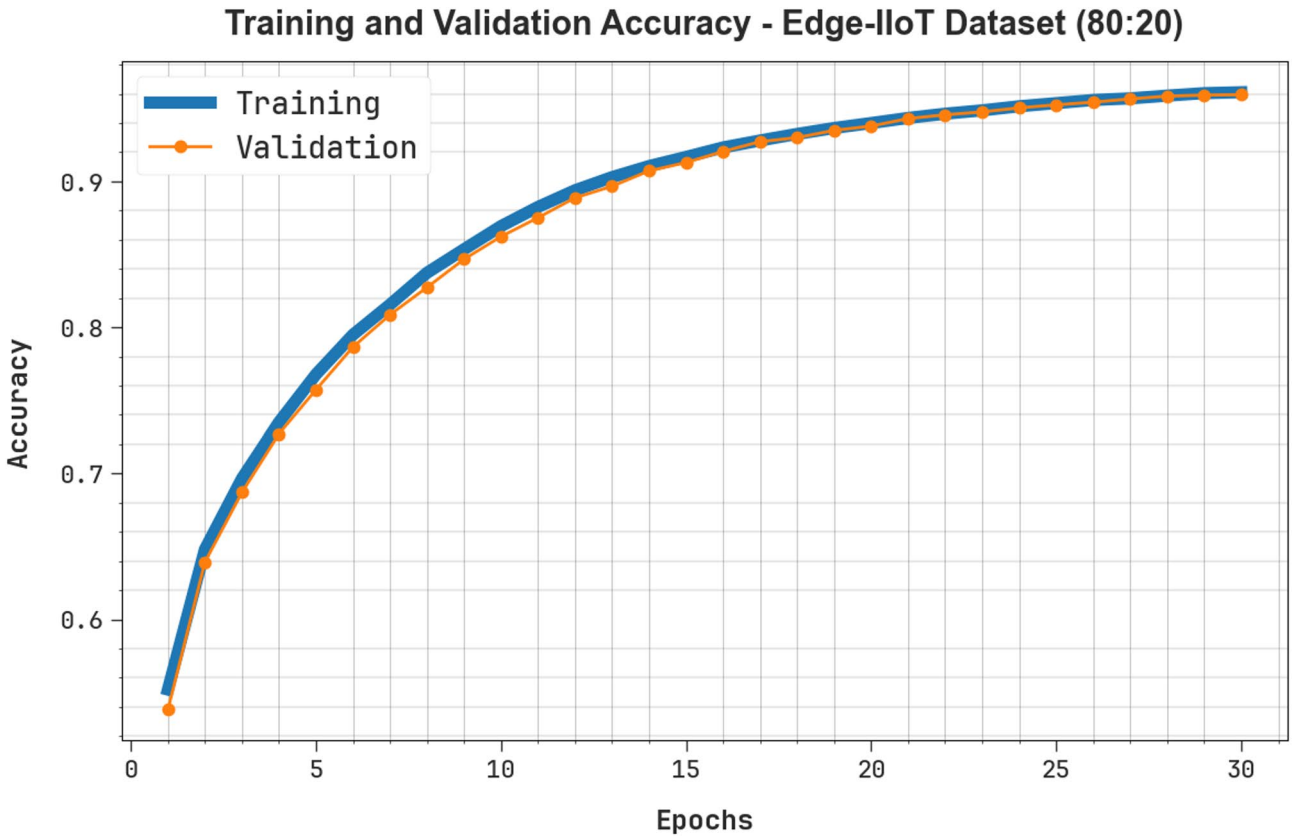

Fig. 8. $Accu_y$ analysis of HDLID-ECSOA method under Edge-IIoT dataset below 80:20.

Also, the proposed HDLID-ECSOA technique is verified under the ToN-IoT dataset[34]. It contains 119,957 samples below nine classes. 42 features are reachable, but only 35 are selected. Table 8 shows the complete particulars of the ToN-IoT dataset.

Figure 14 signifies the confusion matrix created by the HDLID-ECSOA technique under the ToN-IoT dataset. The outcomes specify that the HDLID-ECSOA method efficiently recognizes and detects all classes.

Table 9; Fig. 15 present the intrusion detection of the HDLID-ECSOA methodology under the ToN-IoT dataset. Based on 80% TRAPHA, the proposed HDLID-ECSOA methodology presents an average $accu_y$ of 99.29%, $prec_n$ of 91.52%, $reca_l$ of 86.78%, $F_{Measure}$ of 88.33%, AUC_{Score} of 93.10%, and Kappa of 93.16%. Also, based on 20% TSAPHA, the proposed HDLID-ECSOA technique presents an average $accu_y$ of 99.33%, $prec_n$ of 91.37%, $reca_l$ of 87.07%, $F_{Measure}$ of 88.54%, AUC_{Score} of 93.26%, and Kappa of 93.33%. Besides, depending on 70% TRAPHA, the proposed HDLID-ECSOA technique presents an average $accu_y$ of 99.14%, $prec_n$ of 88.73%, $reca_l$ of 82.60%, $F_{Measure}$ of 83.18%, AUC_{Score} of 90.98%, and Kappa of 91.04%. Finally, based on 30% TSAPHA, the proposed HDLID-ECSOA technique presents an average $accu_y$ of 99.12%, $prec_n$ of 89.48%, $reca_l$ of 82.28%, $F_{Measure}$ of 82.79%, AUC_{Score} of 90.80%, and Kappa of 90.87%.

Figure 16 demonstrates the TRAN $accu_y$ and VALN $accu_y$ results of the HDLID-ECSOA approach under the ToN-IoT dataset below 80:20. The figure highlights that both $accu_y$ values exhibit a growing trend, which indicates the HDLID-ECSOA approach's ability to improve performance across several iterations. Simultaneously, both $accu_y$ remains closer over the epochs, which means lower overfitting and displays greater performance of the HDLID-ECSOA methodology.

Figure 17 shows the TRANLOSS and VALNLOSS analysis of the HDLID-ECSOA approach under the ToN-IoT dataset below 80:20. Both values prove a reducing tendency, informing the capacity of the HDLID-ECSOA model to balance a tradeoff between data fitting and simplification. The incessant decrease in loss values highlights the improved performance of the HDLID-ECSOA model.

The comparative results of the HDLID-ECSOA technique under ToN-IoT dataset with existing techniques are shown in Table 10; Fig. 18[21,35–38]. The results emphasized that the LSTM, RF, AdaBoost, kNN, XGBoost, CART, and 1D CNN techniques have gained minimal performance. However, the proposed HDLID-ECSOA method reported an optimal performance with improved $accu_y$, $prec_n$, $reca_l$ and $F_{Measure}$ of 99.33%, 91.37%, 87.07%, and 88.54%, respectively.

Table 11; Fig. 19 indicates the CT assessment of the HDLID-ECSOA methodology with existing techniques under ToN-IoT dataset. The HDLID-ECSOA methodology attained the fastest CT of 9.69 s, outperforming models like LSTM at 20.08 s, RF at 24.00 s, and EPCOD at 29.67 s. Other approaches such as AdaBoost and DNN recorded even higher CTs of 27.88 s and 28.98 s, respectively. Compared to widely used methods like XGBoost and 1D CNN, which had 12.45 s and 14.87 s respectively, the HDLID-ECSOA model illustrated

Fig. 9. Loss graph of HDLID-ECSOA technique under Edge-IIoT dataset below 80:20.

Edge-IIoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
LSTM	94.83	89.03	89.93	93.99
RF	93.38	92.51	94.63	92.09
FL	96.58	94.43	94.51	89.30
FedMLDL-Bayesian HPO	96.15	91.70	89.02	89.17
LDA Model	98.78	92.51	91.96	91.20
GB	89.04	94.55	89.02	93.23
J48 Method	94.86	95.91	94.93	93.29
CNN-LSTM	94.12	93.28	95.36	92.68
DBN	97.24	94.96	95.29	89.94
NeuroSpatialIOT	96.73	92.24	89.55	89.84
HDLID-ECSOA	99.35	96.11	96.11	96.11

Table 4. Comparative analysis of HDLID-ECSOA model under Edge-IIoT dataset[19,35–38].

superior efficiency. This reduced CT makes the HDLID-ECSOA model highly appropriate for real-time and resource-constrained IIoT environments.

Table 12; Fig. 20 shows the error analysis of the HDLID-ECSOA method with existing models under ToN-IoT dataset. The error analysis on the ToN-IoT dataset reveals that most models showed moderate to low accuracy, with RF achieving 10.47%, AdaBoost 10.08%, and XGBoost 8.05%. These models also exhibited relatively higher $prec_n$, $reca_l$ and $F_{Measure}$ values, such as XGBoost with $prec_n$ of 10.25%, $reca_l$ of 18.79%, and $F_{Measure}$ of 21.39%. In contrast, the HDLID-ECSOA model achieved the lowest $accu_y$ at 0.67%, along with $prec_n$ of 8.63%, $reca_l$ of 12.93%, and $F_{Measure}$ of 11.46%, exhibiting that while it was computationally efficient, its predictive performance was significantly lower. Overall, conventional ensemble methods and tree-based algorithms outperformed others in balancing precision and recall across this dataset.

Table 13; Fig. 21 represent the ablation study of the HDLID-ECSOA approach under the ToN-IoT dataset. The DOA method attains an $accu_y$ of 97.22%, $prec_n$ of 89.47%, $reca_l$ of 84.94%, and an $F_{Measure}$ of 86.42%.

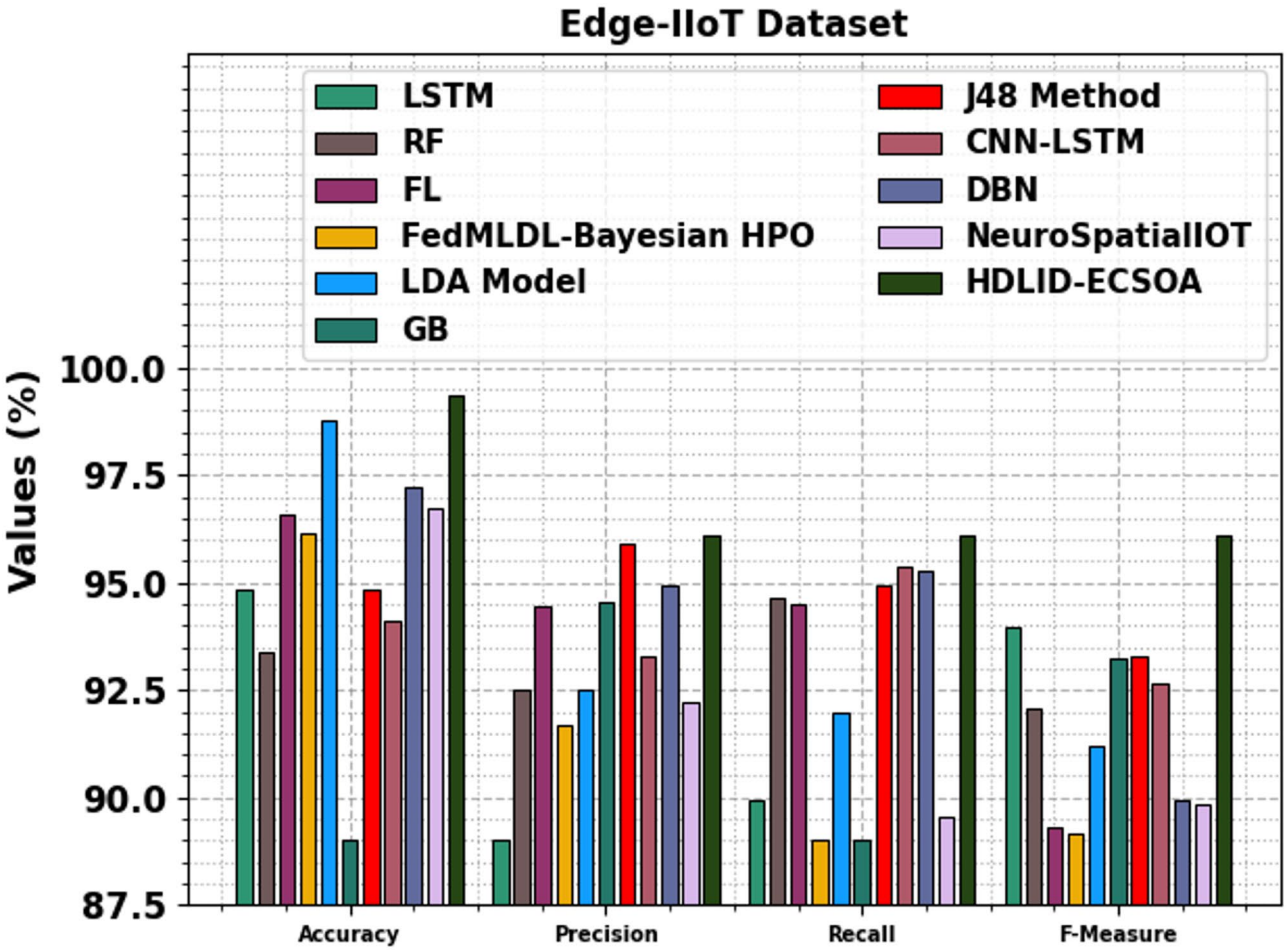

Fig. 10. Comparative analysis of HDLID-ECSOA model under Edge-IIoT dataset.

Edge-IIoT Dataset	
Approach	**CT (sec)**
LSTM	10.93
RF	11.12
FL	15.61
FedMLDL-Bayesian HPO	17.20
LDA Model	11.03
GB	12.39
J48 Method	13.20
CNN-LSTM	15.74
DBN	16.79
NeuroSpatialIOT	12.70
HDLID-ECSOA	7.63

Table 5. CT evaluation of HDLID-ECSOA approach with existing models under Edge-IIoT dataset.

The SFOA approach attains an $accu_y$ of 97.86%, $prec_n$ of 90.02%, $reca_l$ of 85.66%, and an $F_{Measure}$ of 87.18%. The CNN-BiGRU-CRAM model attains an $accu_y$ of 98.53%, $prec_n$ of 90.76%, $reca_l$ of 86.28%, and an $F_{Measure}$ of 87.83%. The proposed HDLID-ECSOA model attains an $accu_y$ of 99.33%, achieving a precision of 91.37%, $reca_l$ of 87.07%, and an $F_{Measure}$ of 88.54%, highlighting its superior detection capability.

Conclusion

In this study, the HDLID-ECSOA technique was presented. The HDLID-ECSOA technique used advanced optimization models to provide intelligent EC in smart cities. Initially, the data pre-processing employed the

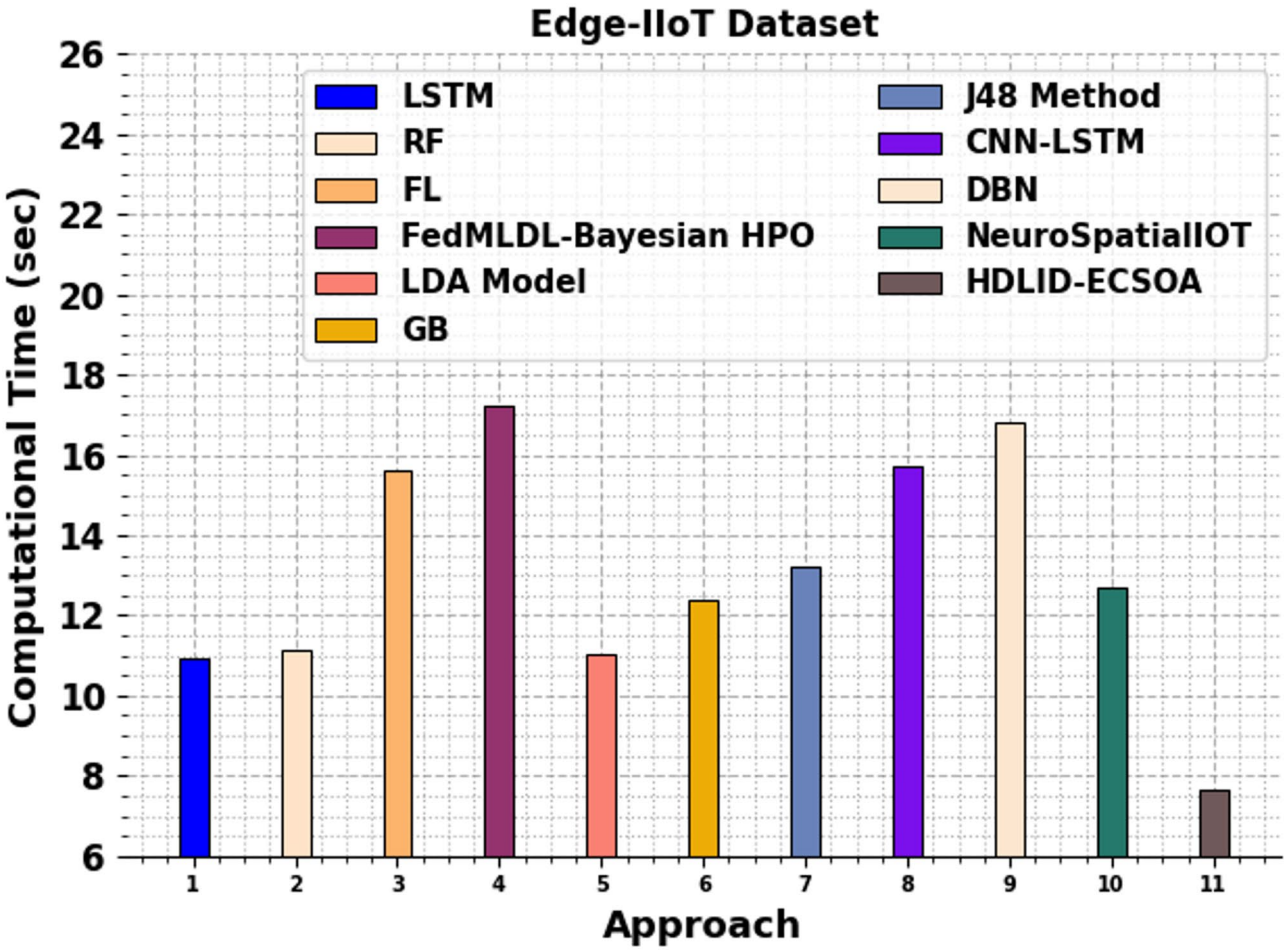

Fig. 11. CT evaluation of HDLID-ECSOA approach with existing models under Edge-IIoT dataset.

Edge-IIoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Recal$	$F_{Measure}$
LSTM	5.17	10.97	10.07	6.01
RF	6.62	7.49	5.37	7.91
FL	3.42	5.57	5.49	10.70
FedMLDL-Bayesian HPO	3.85	8.30	10.98	10.83
LDA Model	1.22	7.49	8.04	8.80
GB	10.96	5.45	10.98	6.77
J48 Method	5.14	4.09	5.07	6.71
CNN-LSTM	5.88	6.72	4.64	7.32
DBN	2.76	5.04	4.71	10.06
NeuroSpatialIOT	3.27	7.76	10.45	10.16
HDLID-ECSOA	0.65	3.89	3.89	3.89

Table 6. Error analysis of HDLID-ECSOA technique with existing methods under Edge-IIoT dataset.

min-max normalization method to convert and standardize raw data to improve the efficiency of models. Furthermore, the DOA implemented a subset of the FS process to detect and choose the most relevant features from the input data. Besides, the HDLID-ECSOA model utilized CNN-BiGRU-CrAM for the classification process. A comprehensive experimentation analysis of the HDLID-ECSOA model is performed under the Edge-IIoT and ToN-IoT datasets. The experimental validation of the HDLID-ECSOA model portrayed superior accuracy values of 99.35% and 99.33% over existing techniques under the dual dataset. The limitations of the HDLID-ECSOA model comprise its focus on a limited set of datasets, which may affect the generalizability of the results to other IoT or IIoT environments. Furthermore, the existing evaluation only considers clean data,

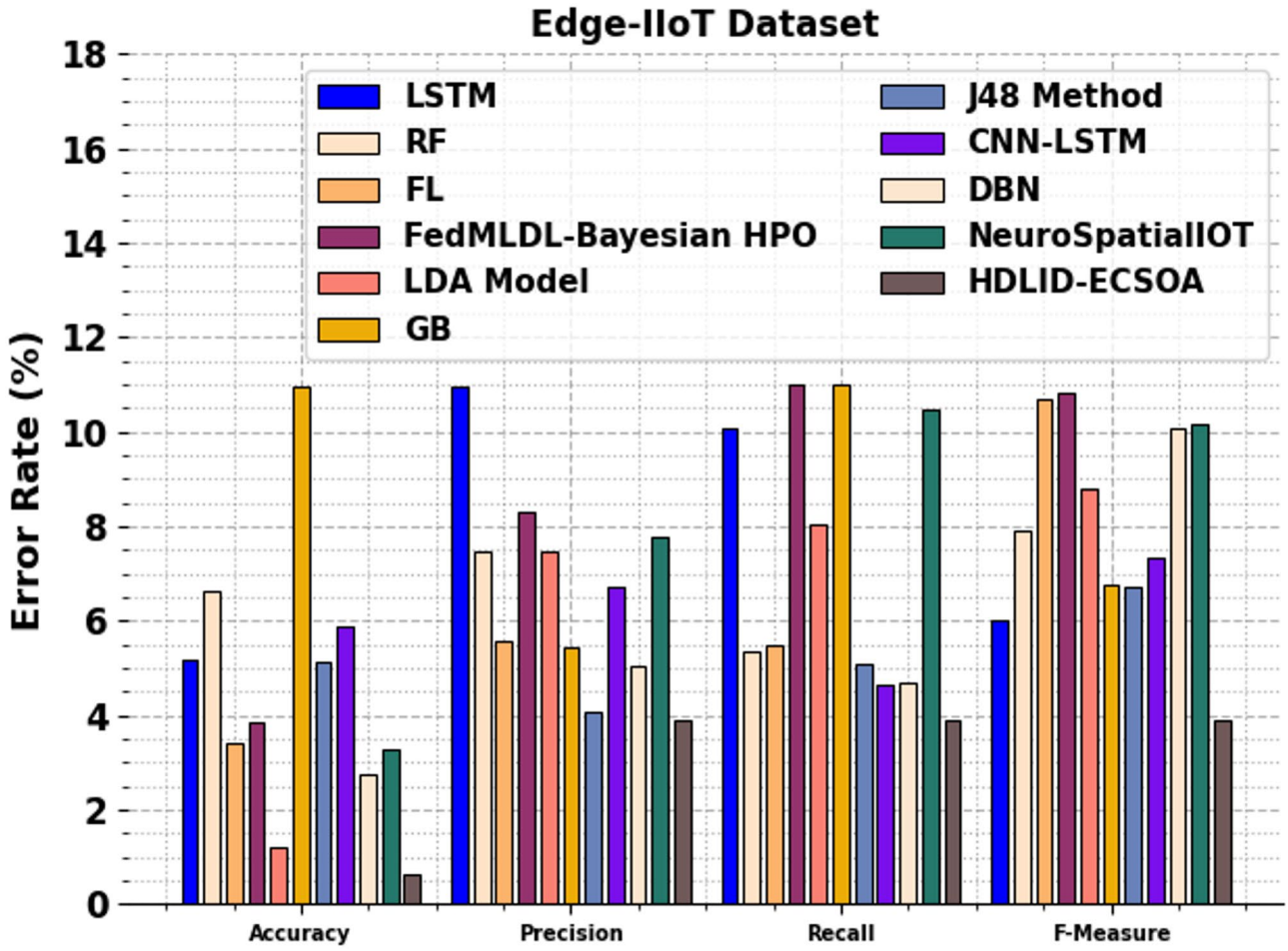

Fig. 12. Error analysis of HDLID-ECSOA technique with existing methods under Edge-IIoT dataset.

Edge-IIoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
DOA	97.23	94.27	94.01	94.14
SFOA	97.75	94.79	94.73	94.69
CNN-BiGRU-CRAM	98.55	95.57	95.41	95.33
HDLID-ECSOA	99.35	96.11	96.11	96.11

Table 7. Result analysis of the ablation study of the HDLID-ECSOA technique.

without examining the robustness of the model against adversarial attacks or noisy data, which are critical in real-world applications. The model does not address privacy and security concerns like data leakage or adversarial evasion strategies. Future work should consider incorporating more diverse datasets, comprising real-world IoT traffic, to validate the performance of the model across diverse scenarios. Additionally, the study may be extended by assessing the model under adversarial conditions to analyze its robustness in hostile environments. Integrating privacy-preserving techniques to enhance security and exploring the scalability of the model in large-scale IoT networks will also be significant areas for future development.

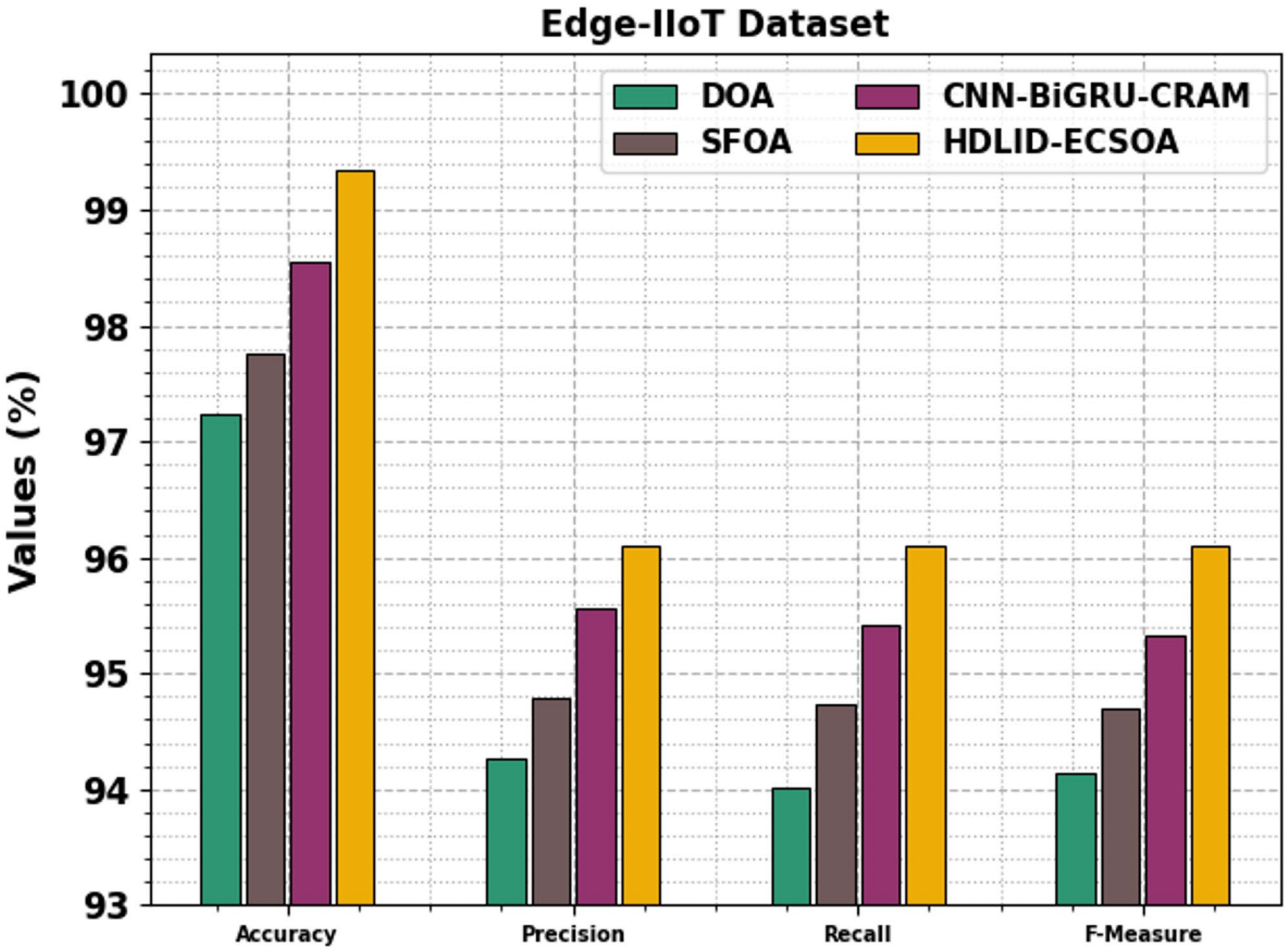

Fig. 13. Result analysis of the ablation study of the HDLID-ECSOA technique.

ToN-IoT Dataset	
Class Labels	**No. of Instances**
"Normal"	78,369
"MiTM"	336
"DoS"	5440
"DDoS"	5987
"Password"	6016
"Injection"	5867
"XSS"	5951
"Ransomware"	5976
"Backdoor"	6015
Total Instances	**119,957**

Table 8. Details of the ToN-IoT dataset.

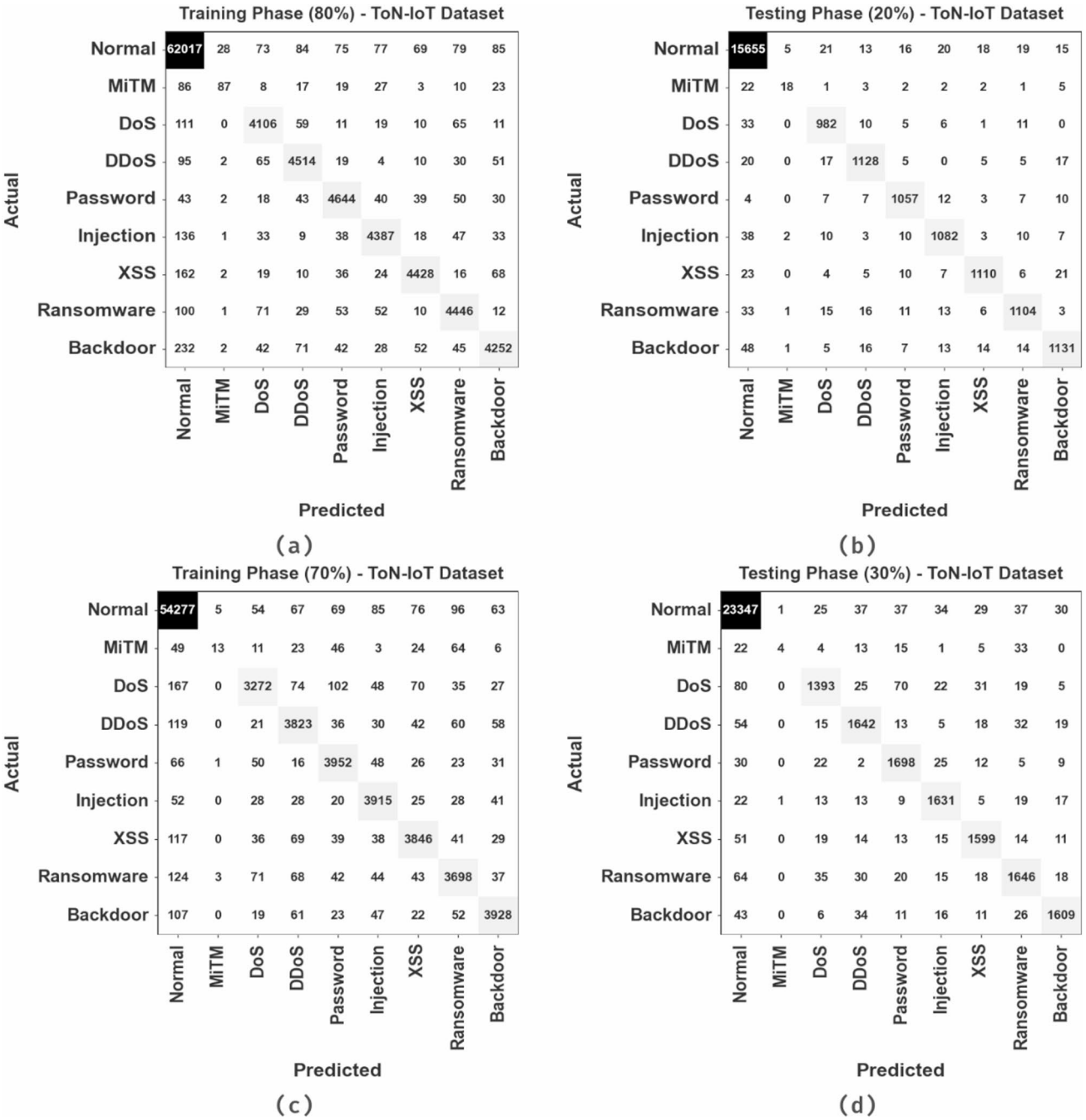

Fig. 14. ToN-IoT dataset (a-b) 80%TRAPHA and 20%TESPHA and (c-d) 70%TRAPHA and 30%TESPHA.

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$	AUC_{Score}	Kappa
TRAPHA (80%)						
Normal	98.40	98.47	99.09	98.78	98.10	98.15
MiTM	99.76	69.60	31.07	42.96	65.52	65.57
DoS	99.36	92.58	93.49	93.03	96.56	96.62
DDoS	99.38	93.34	94.24	93.79	96.94	97.00
Password	99.42	94.07	94.60	94.33	97.14	97.20
Injection	99.39	94.18	93.30	93.74	96.50	96.56
XSS	99.43	95.45	92.93	94.17	96.35	96.40
Ransomware	99.30	92.86	93.13	92.99	96.38	96.45
Backdoor	99.14	93.14	89.22	91.14	94.44	94.50
Average	99.29	91.52	86.78	88.33	93.10	93.16
TESPHA (20%)						
Normal	98.55	98.61	99.20	98.90	98.25	98.33
MiTM	99.80	66.67	32.14	43.37	66.05	66.13
DoS	99.39	92.47	93.70	93.08	96.68	96.73
DDoS	99.41	93.92	94.24	94.08	96.96	97.04
Password	99.52	94.12	95.48	94.80	97.60	97.66
Injection	99.35	93.68	92.88	93.28	96.28	96.35
XSS	99.47	95.52	93.59	94.55	96.68	96.73
Ransomware	99.29	93.80	91.85	92.81	95.76	95.82
Backdoor	99.18	93.55	90.55	92.03	95.10	95.17
Average	99.33	91.37	87.07	88.54	93.26	93.33
TRAPHA (70%)						
Normal	98.43	98.55	99.06	98.80	98.16	98.23
MiTM	99.72	59.09	05.44	09.96	52.71	52.78
DoS	99.03	91.86	86.22	88.95	92.93	92.99
DDoS	99.08	90.40	91.26	90.83	95.38	95.45
Password	99.24	91.29	93.80	92.53	96.67	96.73
Injection	99.33	91.94	94.63	93.27	97.10	97.17
XSS	99.17	92.14	91.25	91.69	95.42	95.48
Ransomware	99.01	90.26	89.54	89.90	94.52	94.57
Backdoor	99.26	93.08	92.23	92.65	95.93	96.01
Average	99.14	88.73	82.60	83.18	90.98	91.04
TESPHA (30%)						
Normal	98.34	98.46	99.02	98.74	98.04	98.10
MiTM	99.74	66.67	04.12	07.77	52.06	52.12
DoS	98.91	90.93	84.68	87.69	92.14	92.21
DDoS	99.10	90.72	91.32	91.02	95.42	95.47
Password	99.19	90.03	94.18	92.06	96.81	96.88
Injection	99.36	92.46	94.28	93.36	96.94	97.02
XSS	99.26	92.53	92.11	92.32	95.87	95.94
Ransomware	98.93	89.90	89.17	89.53	94.31	94.38
Backdoor	99.29	93.66	91.63	92.63	95.66	95.73
Average	99.12	89.48	82.28	82.79	90.80	90.87

Table 9. Intrusion detection of HDLID-ECSOA approach under ToN-IoT dataset.

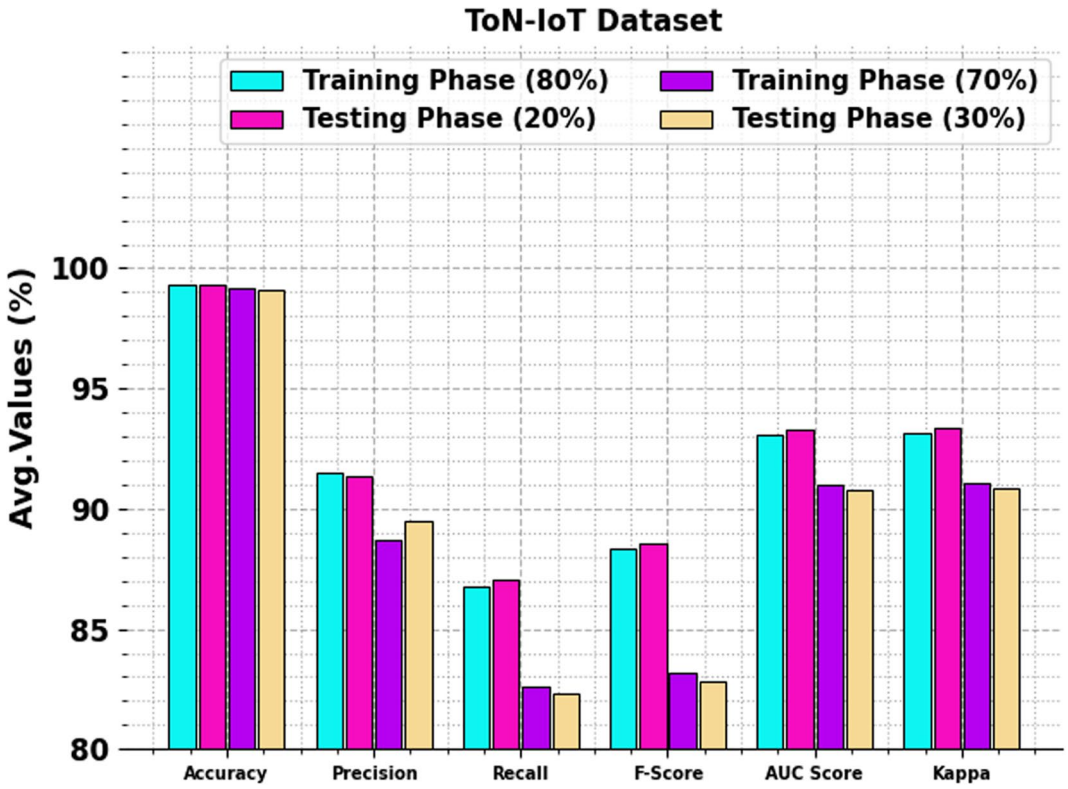

Fig. 15. Average of HDLID-ECSOA approach under ToN-IoT dataset.

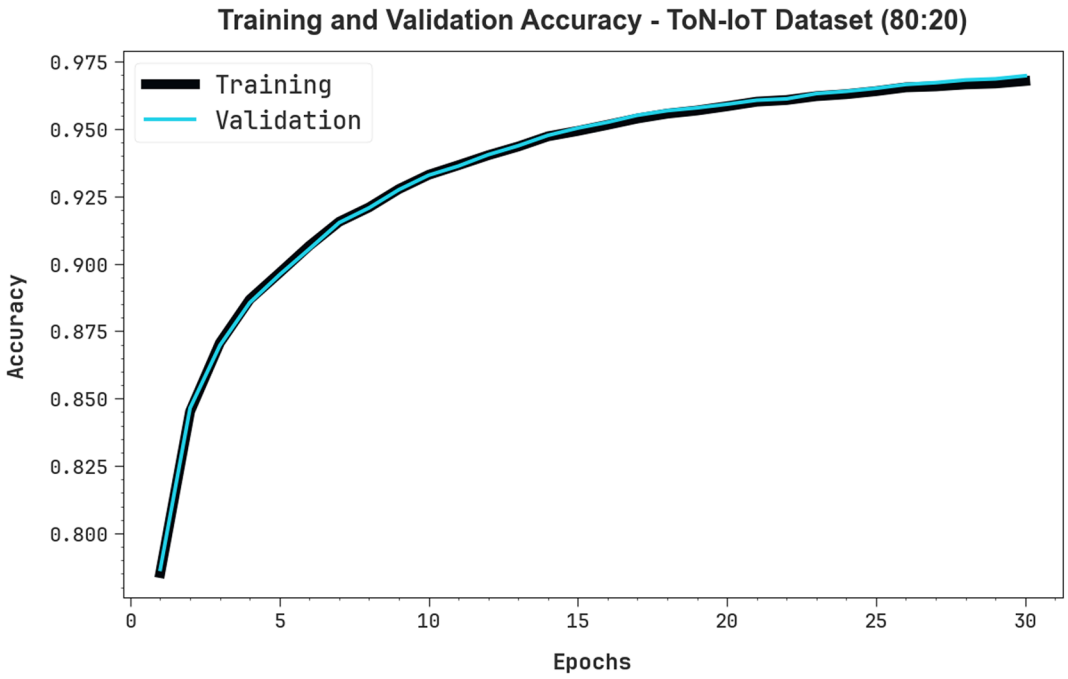

Fig. 16. $Accu_y$ analysis of HDLID-ECSOA approach under ToN-IoT dataset below 80:20.

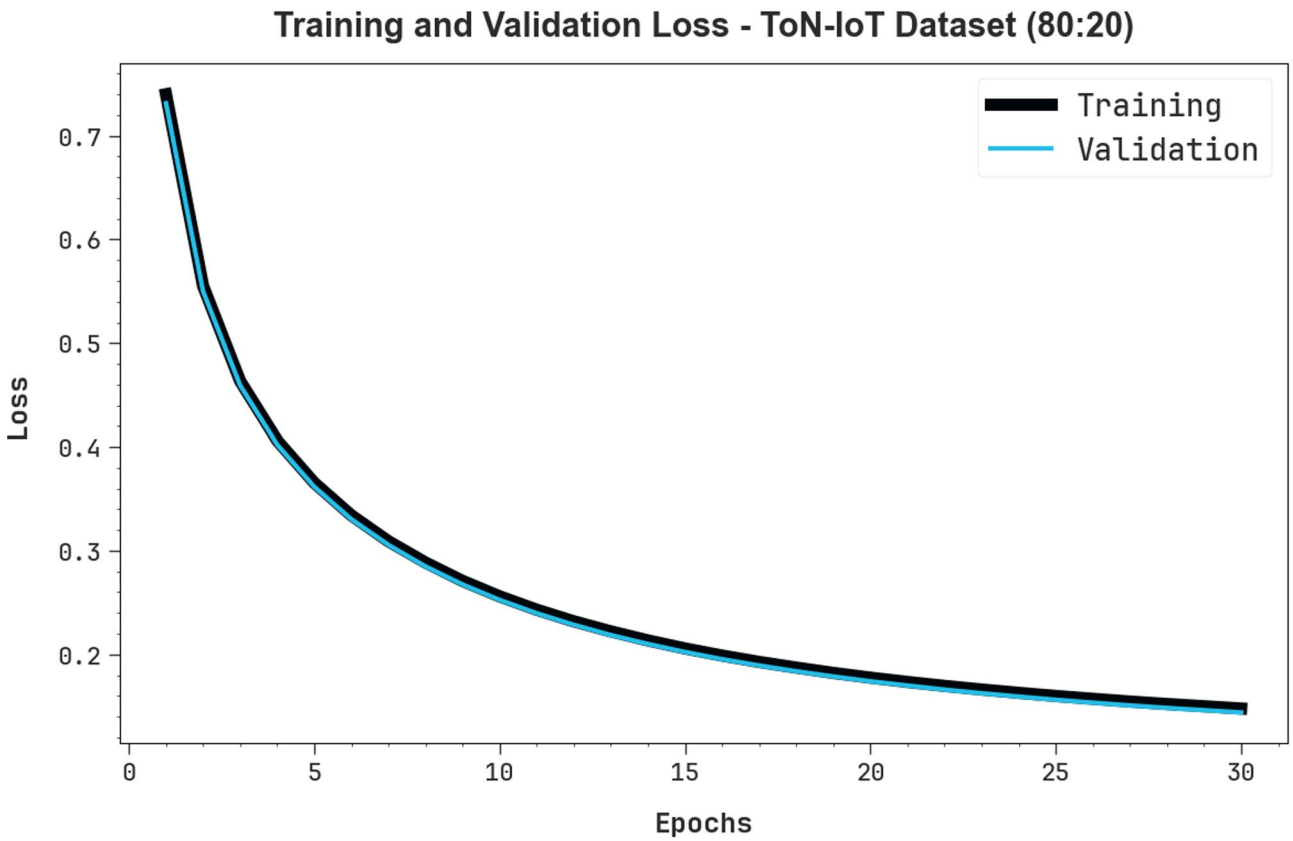

Fig. 17. Loss graph of HDLID-ECSOA approach under ToN-IoT dataset below 80:20.

ToN-IoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
LSTM	97.67	89.96	81.23	81.89
RF	89.53	89.92	86.14	80.11
AdaBoost	89.92	89.66	79.65	84.61
kNN Algorithm	94.58	89.89	80.25	82.15
XGBoost	91.95	89.75	81.21	78.61
CART Method	95.89	89.30	83.58	80.63
1D CNN	97.46	89.87	82.86	84.56
EPCOD	90.15	90.57	86.90	80.88
DNN	90.42	90.33	80.18	85.15
EEDOS	95.29	90.41	80.83	82.87
HDLID-ECSOA	99.33	91.37	87.07	88.54

Table 10. Comparative analysis of HDLID-ECSOA model under ToN-IoT dataset[21,35–38].

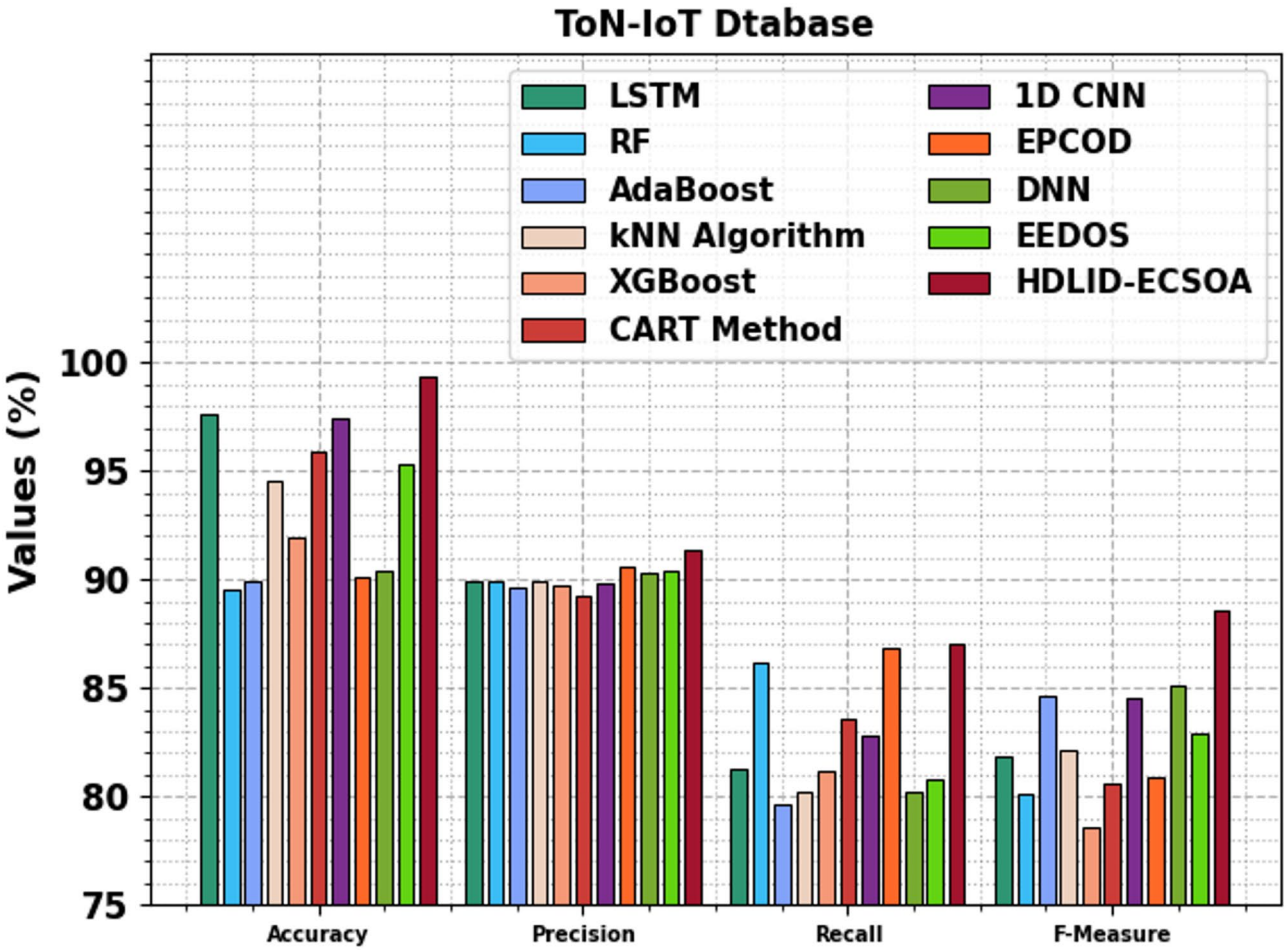

Fig. 18. Comparative analysis of HDLID-ECSOA model under ToN-IoT dataset.

ToN-IoT Dataset	
Approach	**CT (sec)**
LSTM	20.08
RF	24.00
AdaBoost	27.88
kNN Algorithm	17.59
XGBoost	12.45
CART Method	23.38
1D CNN	14.87
EPCOD	29.67
DNN	28.98
EEDOS	17.98
HDLID-ECSOA	9.69

Table 11. CT assessment of HDLID-ECSOA methodology with existing techniques under ToN-IoT dataset.

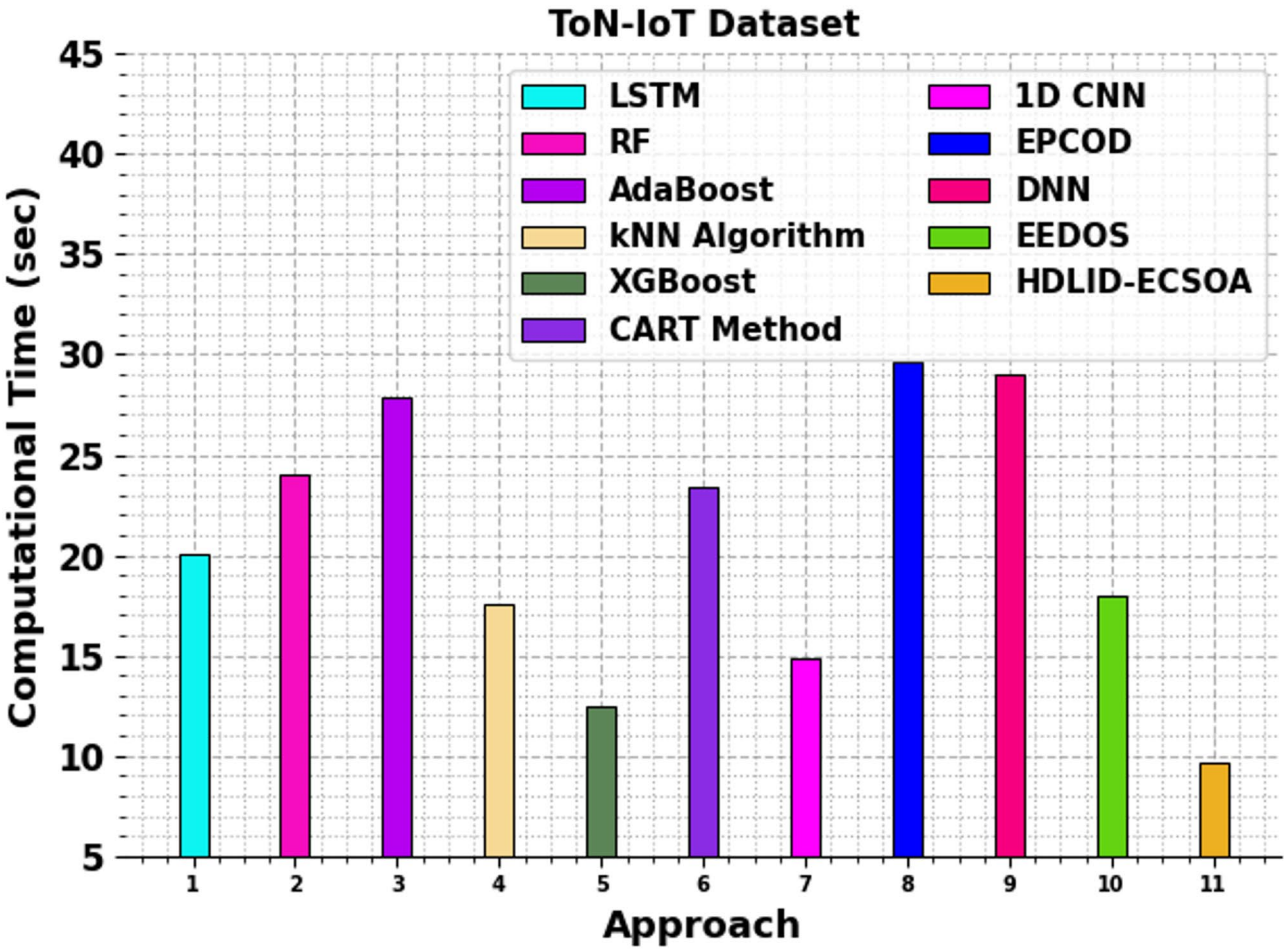

Fig. 19. CT assessment of HDLID-ECSOA methodology with existing techniques under ToN-IoT dataset.

ToN-IoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
LSTM	2.33	10.04	18.77	18.11
RF	10.47	10.08	13.86	19.89
AdaBoost	10.08	10.34	20.35	15.39
kNN Algorithm	5.42	10.11	19.75	17.85
XGBoost	8.05	10.25	18.79	21.39
CART Method	4.11	10.70	16.42	19.37
1D CNN	2.54	10.13	17.14	15.44
EPCOD	9.85	9.43	13.10	19.12
DNN	9.58	9.67	19.82	14.85
EEDOS	4.71	9.59	19.17	17.13
HDLID-ECSOA	0.67	8.63	12.93	11.46

Table 12. Error analysis of HDLID-ECSOA method with existing models under ToN-IoT dataset.

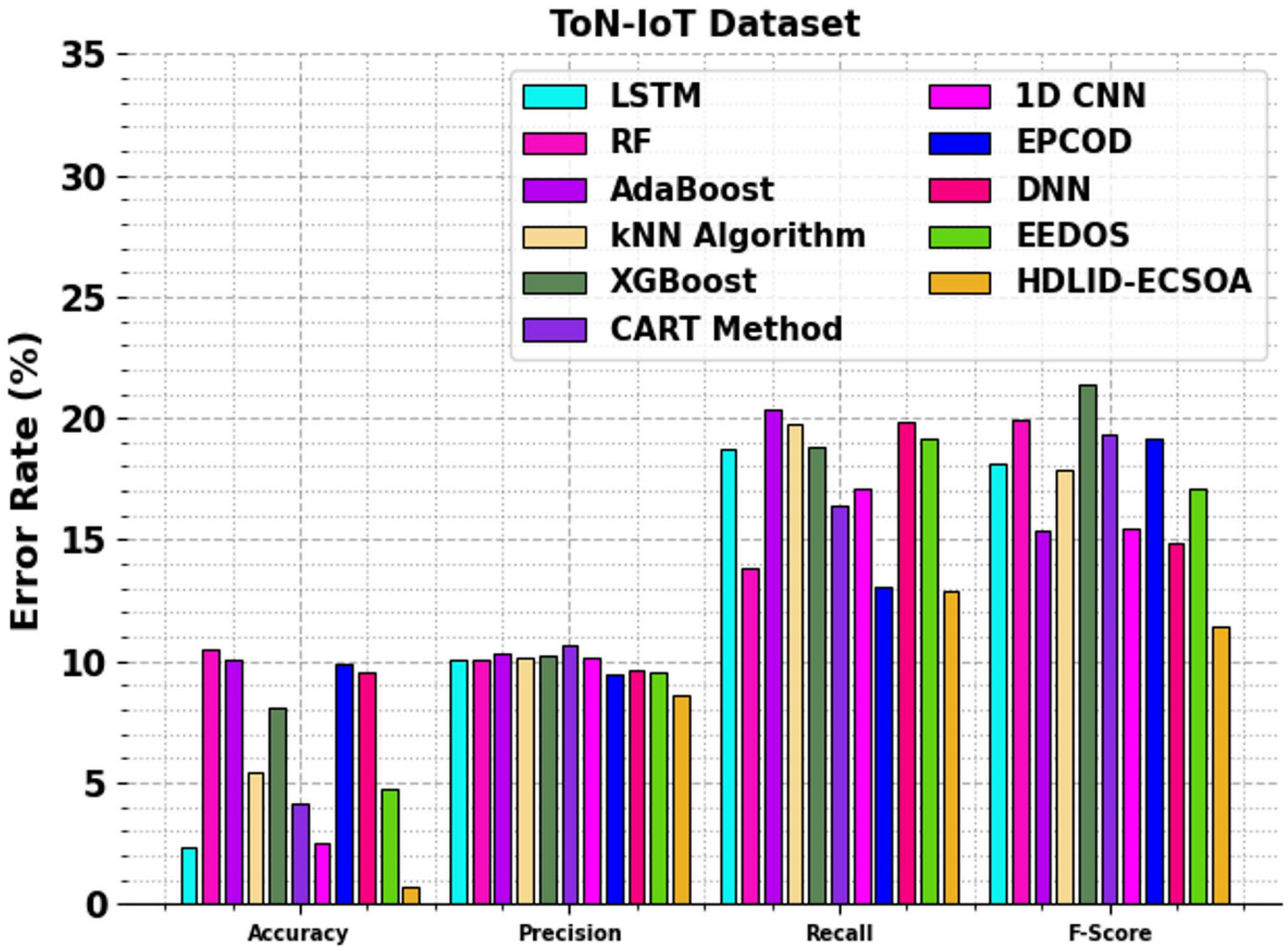

Fig. 20. Error analysis of HDLID-ECSOA method with existing models under ToN-IoT dataset.

ToN-IoT Dataset				
Approach	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
DOA	97.22	89.47	84.94	86.42
SFOA	97.86	90.02	85.66	87.18
CNN-BiGRU-CRAM	98.53	90.76	86.28	87.83
HDLID-ECSOA	99.33	91.37	87.07	88.54

Table 13. Result analysis of the ablation study of the HDLID-ECSOA approach.

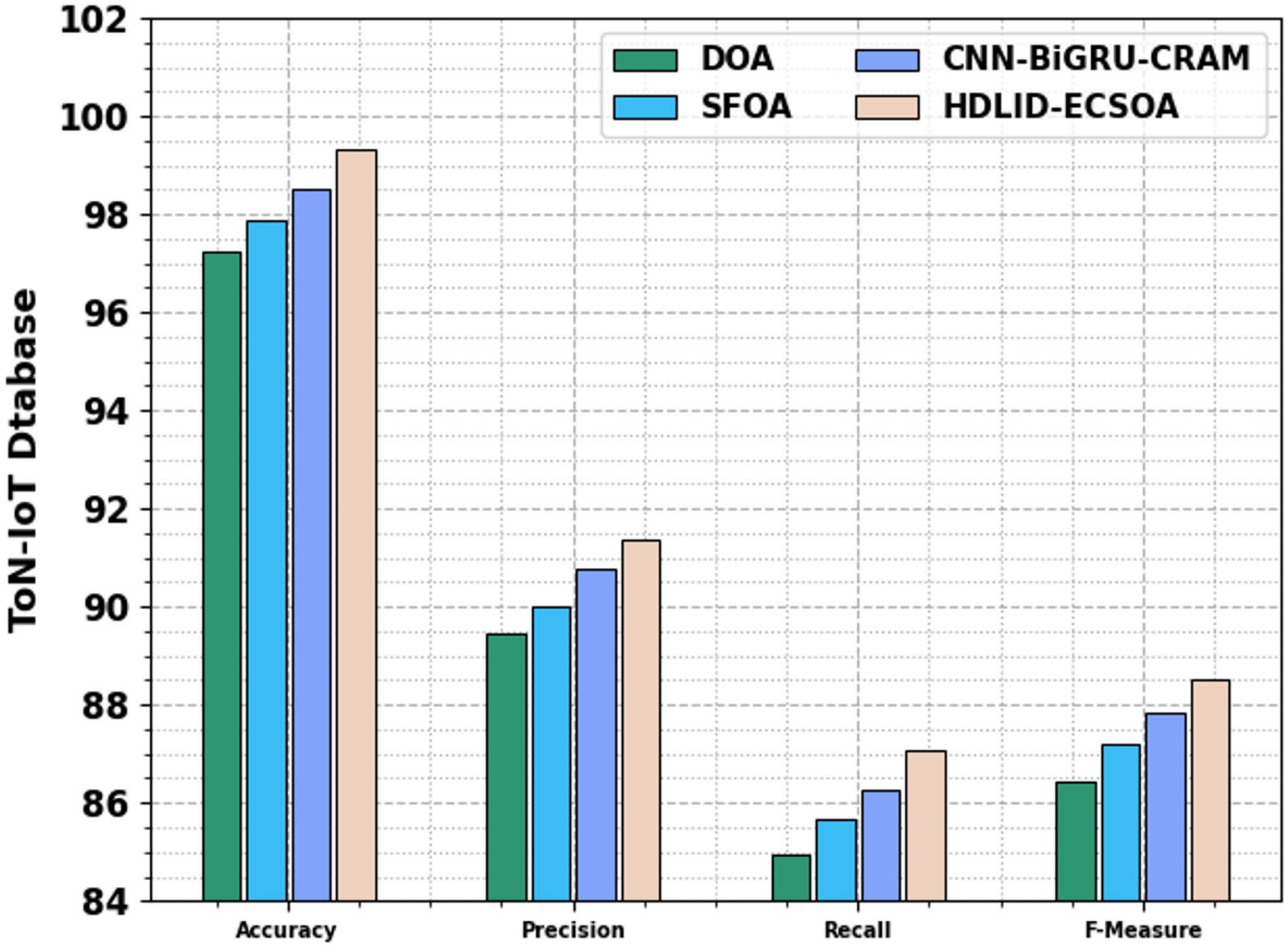

Fig. 21. Result analysis of the ablation study of the HDLID-ECSOA approach.

Data availability

The data that support this study's findings are openly available in the Kaggle repository at https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot and https://www.kaggle.com/datasets/dhoogla/cictoniot, reference numbers[33,34].

Received: 16 March 2025; Accepted: 11 July 2025
Published online: 26 September 2025

References

1. Sumathi, S., Pawar, S. & L, S. K. B. Hybrid chaotic Bat artificial bee colony algorithm assisted hybrid machine learning based intrusion detection system. *Fusion Pract. Appl.* **19** (2), 45–63. https://doi.org/10.54216/fpa.190204 (2025).
2. Mohy-Eddine, M., Guezzaz, A., Benkirane, S. & Azrour, M. A practical intrusion detection approach based on ensemble learning for IIoT edge computing. *J. Comput. Virol. Hacking Techniques*. **19** (4), 469–481 (2023).
3. Osman, L., Taiwo, O., Elashry, A. & Ezugwu, A. E. Intelligent edge computing for iot: enhancing security and privacy. *J. Intell. Syst. Internet Things* **8**(1) (2023).
4. Yang, R. et al. Efficient intrusion detection toward IoT networks using cloud–edge collaboration. *Computer Networks*, *228*, p.109724. (2023).
5. Gyamfi, E. & Jurcut, A. Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors*, *22*(10), p.3744. (2022).
6. Singh, A., Chatterjee, K. & Satapathy, S. C. An edge based hybrid intrusion detection framework for mobile edge computing. *Complex. Intell. Syst.* **8** (5), 3719–3746 (2022).
7. Haq, M. A., Rahim Khan, M. A. & AL-Harbi, T. Development of PCCNN-based network intrusion detection system for EDGE computing. *Comput. Mater. Continua* **71**(1). (2022).
8. Alzubi, O. A. et al. Optimized machine learning-based intrusion detection system for fog and edge computing environment. *Electronics*, *11*(19), p.3007. (2022).
9. Man, D. et al. Intelligent Intrusion Detection Based on Federated Learning for Edge-Assisted Internet of Things. *Security and Communication Networks*, *2021*(1), p.9361348. (2021).
10. Spadaccino, P. & Cuomo, F. Intrusion detection systems for iot: opportunities and challenges offered by edge computing and machine learning. *arXiv preprint arXiv:2012.01174*. (2020).
11. Chen, Y., Ding, Y., Hu, Z. Z. & Ren, Z. Geometrized task scheduling and adaptive resource allocation for Large-Scale edge computing in smart cities. *IEEE Internet Things J.* (2025).

12. Al-Quayed, F. et al. Context-Aware prediction with secure and lightweight cognitive decision model in smart cities. *Cogn. Comput.* **17** (1), 1–12 (2025).
13. Sahu, D. et al. A multi objective optimization framework for smart parking using digital twin pareto front MDP and PSO for smart cities. *Scientific Reports*, *15*(1), p.7783. (2025).
14. Chen, D., Wang, S., Wang, C., Zhang, X. & Chen, N. Enhanced sensor web services by incorporating IoT interface protocols and spatio-temporal data streams for edge computing-based sensing. *Geo-spatial Inform. Sci.*, pp.1–18. (2025).
15. Wang, M. et al. Smart City transportation: A VANET edge computing model to minimize latency and delay utilizing 5G network. *J. Grid Comput.* **22** (1), 25 (2024).
16. Tian, H., Li, R., Di, Y., Zuo, Q. & Wang, J. Employing RNN and Petri Nets to Secure Edge Computing Threats in Smart Cities. *Journal of Grid Computing*, *22*(1), p.32. (2024).
17. Xu, R., Nagothu, D. & Chen, Y. AR-Edge: Autonomous and Resilient Edge Computing Architecture for Smart Cities. In *Edge Computing Architecture-Architecture and Applications for Smart Cities*. IntechOpen. (2024).
18. Sun, Z. et al. A resource allocation scheme for edge computing network in smart City based on attention mechanism. *ACM Trans. Sens. Networks* (2024).
19. Wang, W., Wang, K. & Du, H. Design and optimization of human-machine interaction interface for the intelligent Internet of Things based on deep learning and spatial computing. *Egyptian Informatics Journal*, *30*, p.100685. (2025).
20. Far, A. Z. et al. Artificial intelligence for secured information systems in smart cities: Collaborative iot computing with deep reinforcement learning and blockchain. *arXiv preprint arXiv:2409.16444*. (2024).
21. Khan, H. et al. A deep learning-based strategy for energy-efficient parallel computation offloading in mobile edge networks. *Ad Hoc Networks*, p.103787. (2025).
22. Mishra, S. & Chaurasiya, V. K. Hybrid deep learning algorithm for smart cities security enhancement through blockchain and internet of things. *Multimedia Tools Appl.* **83** (8), 22609–22637 (2024).
23. Ficili, I., Giacobbe, M., Tricomi, G. & Puliafito, A. From sensors to data intelligence: Leveraging IoT, cloud, and edge computing with AI. *Sensors*, *25*(6), p.1763. (2025).
24. Wang, B., Dabbaghjamanesh, M., Kavousi-Fard, A. & Yue, Y. AI-enhanced multi-stage learning-to-learning approach for secure smart cities load management in IoT networks. *Ad Hoc Networks*, *164*, p.103628. (2024).
25. Lilhore, U. K. et al. Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. *J. Cloud Comput.* **14** (1), 5 (2025).
26. Ahmed, K. & Elena, P. Integrating artificial intelligence with edge computing for scalable autonomous networks. *Am. J. Technol. Advancement*. **1** (8), 57–81 (2024).
27. Al-Zaidawi, Q. J. & Çevik, M. M. and Advanced Deep Learning Models for Improved IoT Network Monitoring Using Hybrid Optimization and MCDM Techniques. *Symmetry*, *17*(3), p.388. (2025).
28. Kumar, P. J. & Nedunchelivan, S. A Shark Inspired Ensemble Deep Learning Stacks for Ensuring the Security in Internet of Things (IoT)-Based Smart City Infrastructure. *International Journal of Computational Intelligence Systems*, *17*(1), p.243. (2024).
29. Tiwari, D. et al. A swarm-optimization based fusion model of sentiment analysis for cryptocurrency price prediction. *Scientific Reports*, *15*(1), p.8119. (2025).
30. Houssein, E. H., Hossam Abdel Gafar, M., Fawzy, N. & Sayed, A. Y. Recent metaheuristic algorithms for solving some civil engineering optimization problems. *Scientific Reports*, *15*(1), p.7929. (2025).
31. Logapriya, E., Rajendran, S. & Zakariah, M. Hybrid Greylag Goose deep learning with layered sparse network for women nutrition recommendation during menstrual cycle. *Scientific Reports*, *15*(1), p.5959. (2025).
32. Dey, R. et al. A Hybrid Evolutionary Fuzzy Ensemble Approach for Accurate Software Defect Prediction. (2025).
33. https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot
34. https://www.kaggle.com/datasets/dhoogla/cictoniot
35. Chandnani, C. J. et al. A physics based hyper parameter optimized federated Multi-Layered deep learning model for intrusion detection in IoT networks. *IEEE Access* (2025).
36. Gad, A. R., Nashat, A. A. & Barkat, T. M. Intrusion detection system using machine learning for vehicular ad hoc networks based under ToN-IoT dataset. *IEEE Access* **9**, 142206–142217 (2021).
37. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. & Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *Ieee Access.* **8**, 165130–165150 (2020).
38. Olawale, O. P. & Ebadinezhad, S. Cybersecurity anomaly detection: Ai and Ethereum blockchain for a secure and tamperproof Ioht data management. *IEEE Access* (2024).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/231/46. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R721), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Ongoing Research Funding program, (ORF-2025-459), King Saud University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2025-1661-06". The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

Author contributions

Amal K. Alkhalifa: Conceptualization, methodology development, experiment, formal analysis, investigation, writing. Mohammed Aljebreen: Formal analysis, investigation, validation, visualization, writing. Nazir Ahmad: Formal analysis, review and editing. Othman Alrusaini: Methodology, investigation. Nojood O Aljehane: Review and editing.Ali Alqazzaz: Discussion, review and editing. Hassan Alkhiri: Discussion, review and editing. Rakan Alanazi: Conceptualization, methodology development, investigation, supervision, review and editing. All authors have read and agreed to the published version of the manuscript.

Declarations

Conflict of interest

The authors declare that they have no conflict of interest. The manuscript was written with the contributions of all authors, and all authors have approved the final version.

Ethics approval
This article contains no studies with human participants performed by any authors.

Consent to participate
Not applicable.

Informed consent
Not applicable.

Additional information
Correspondence and requests for materials should be addressed to R.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.