scientific reports

Check for updates

OPEN

Modeling of Bayesian machine learning with sparrow search algorithm for cyberattack detection in IIoT environment

Faten Khalid Karim[1], José Varela-Aldás[2]✉, Mohamad Khairi Ishak[3], Ayman Aljarbouh[4] & Samih M. Mostafa[5,6]

With the fast-growing interconnection of smart technologies, the Industrial Internet of Things (IIoT) has revolutionized how industries work by connecting devices and sensors and automating regular operations via the Internet of Things (IoTs). IoT devices provide seamless diversity and connectivity in different application domains. This system and its transmission channels are subjected to targeted cyberattacks due to their round-the-clock connectivity. Accordingly, a multilevel security solution is needed to safeguard the industrial system. By analyzing the data packet, the Intrusion Detection System (IDS) counteracts the cyberattack for the targeted attack in the IIoT platform. Various research has been undertaken to address the concerns of cyberattacks on IIoT networks using machine learning (ML) and deep learning (DL) approaches. This study introduces a new Bayesian Machine Learning with the Sparrow Search Algorithm for Cyberattack Detection (BMLSSA-CAD) technique in the IIoT networks. The proposed BMLSSA-CAD technique aims to enhance security in IIoT networks by detecting cyberattacks. In the BMLSSA-CAD technique, the min-max scaler normalizes the input dataset. Additionally, the method utilizes the Chameleon Optimization Algorithm (COA)-based feature selection (FS) approach to identify the optimal feature set. The BMLSSA-CAD technique uses the Bayesian Belief Network (BBN) model for cyberattack detection. The hyperparameter tuning process employs the sparrow search algorithm (SSA) model to enhance the BBN model performance. The performance of the BMLSSA-CAD method is examined using UNSWNB51 and UCI SECOM datasets. The experimental validation of the BMLSSA-CAD method highlighted superior accuracy outcomes of 97.84% and 98.93% compared to recent techniques on the IIoT platform.

Keywords Industrial internet of things, Cyberattack detection, Bayesian machine learning, Chameleon optimization algorithm, Sparrow search algorithm

The emergence of the Internet of Things (IoT) paradigm in Industrial Automation and Control Systems (IACS) is named Industrial IoT (IIoT), which currently has become very famous[1]. The IACS has been used recently to retain an eye on manufacturing machines and methods, so the IIoT-based systems have become a vital part of each crucial infrastructure in smart industries. The significant portions of these methods are the data acquisition and supervisory methods that frequently control the IACSs[2]. Real monitoring, contact with the devices, data analysis, and logging of every event in the methods are the foremost parts of these systems. Therefore, the arrival of the IoT model in these systems improves the safety and network intellect in the optimization and computerization of industrial methods[3]. Industrial Internet of Things (IIoT) systems handle vast amounts of data, and their applications are often mission-critical and require high availability. It is, therefore, essential to implement robust cybersecurity measures to protect these systems adequately. In data management, cybersecurity has become

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, 11671 Riyadh, Saudi Arabia. [2]Centro de Investigación en Ciencias Humanas y de la Educación - CICHE, Facultad de Ingenierías, Ingeniería industrial, Universidad Tecnológica Indoamérica, Ambato, Ecuador. [3]Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates. [4]Department of Computer Science, School of Arts and Sciences, University of Central Asia, Naryn, Kyrgyzstan. [5]Computer Science Department, Faculty of Computers and Information, South Valley University, Qena 83523, Egypt. [6]Faculty of Industry and Energy Technology, New Assiut Technological University (N.A.T.U.), New Asyut 71684, Egypt. ✉email: josevarela@uti.edu.ec

indispensable in the current Internet of Things (IoT) environment[4]. The widespread adoption of IoT devices in homes, the integration of smart cars and smart power grids, and the complexity of communication protocols used by IoT consumers have significantly increased the exposure of IoT systems to cyber-attacks. Assaults can arise through various physical and cyber methods, and both can happen in smart industries and cities[5]. These attacks contain a permanent denial of service (DoS), side-channel, sleep denial attacks, malicious code injection, radio frequency blocking, and false node injection. In a cyber-attack, the attacker initially tries to obtain illegal access to the system modules by inserting dangerous software like malware into the devices[6].

These types of attacks contain distributed DoS attacks (DDoS), ransomware, and man-in-the-middle attacks (MITM). Only some models (like signature base) were developed in the literature to resolve the problem. In the signature-based model, a group of attacks is tested beside the present doubtful models. If the signature extractor technique is not completely capable of taking the separate feature of attacks, it may mainly generate false alarms or misdetection of attacks[7]. This system is not appropriate for classifying unknown assaults and undergoes excellent handling overhead. Machine learning (ML) models can identify assaults at execution time and consume less processing time than other models. The application of numerous DL models can recognize assaults with dual identification and categorize dissimilar types of attacks using multiclass classification, which is an active study domain[8]. While numerous extensive analyses have explored this emerging field of study, the survey needs to provide a balanced comparison of various deep learning approaches, particularly in applying novel datasets for intrusion detection. The rapid expansion of interconnected devices within IIoT environments also intensifies the potential susceptibilities to cyber threats[9]. Protecting these systems from malevolent attacks is paramount to conserving significant infrastructure, averting potential disruptions, and preserving data integrity. This study uses novel models to innovate cyberattack recognition methodologies in IIoT settings. By improving the resilience and security posture of IIoT systems, this study aims to safeguard the reliability and safety of industrial procedures in the face of growing cybersecurity challenges[10].

This study introduces a new Bayesian Machine Learning with the Sparrow Search Algorithm for Cyberattack Detection (BMLSSA-CAD) technique in the IIoT networks. The proposed BMLSSA-CAD technique aims to enhance security in IIoT networks by detecting cyberattacks. In the BMLSSA-CAD technique, the min-max scaler normalizes the input dataset. Additionally, the method utilizes the Chameleon Optimization Algorithm (COA)-based feature selection (FS) approach to identify the optimal feature set. The BMLSSA-CAD technique uses the Bayesian Belief Network (BBN) model for cyberattack detection. The hyperparameter tuning process employs the sparrow search algorithm (SSA) model to enhance the BBN model performance. The performance of the BMLSSA-CAD method is examined using UNSWNB51 and UCI SECOM datasets. The significant contribution of the BMLSSA-CAD method is as follows:

- The BMLSSA-CAD approach utilizes min-max scaling to normalize the input dataset, significantly safeguarding uniform feature ranges to improve model convergence and training stability. This step contributes to an enhanced predictive accomplishment by reducing the impact of varying data scales, facilitating more efficient learning and accurate cyberattack recognition within diverse and dynamic datasets.
- The COA method is employed for feature selection, crucial in improving cyberattack detection by detecting the most relevant features. This technique mitigates dimensionality, enhancing computational efficiency and confirming that the model concentrates on critical indicators of cyber threats. By optimizing feature sets dynamically, COA contributes crucially to the accuracy and efficiency of the model in recognizing and reducing various cybersecurity risks.
- The BMLSSA-CAD model utilizes BBN to model dependencies among features, giving a probabilistic framework that improves cyberattack recognition and classification. By comprehending complex associations between variables, BBNs enable the method to conclude potential cyber threats more precisely, enhancing overall recognition performance. This method contributes to robust cybersecurity by giving a structured methodology to analyze and respond to growing attack patterns based on probabilistic reasoning and feature interactions.
- The SSA is applied to tune the BMLSSA-CAD technique, which is central in optimizing model parameters to improve BBN performance. By systematically altering hyperparameters, SSA enhances the BBN's ability to recognize and reduce cyber threats effectively. This approach contributes to the model's adaptability and efficiency in dynamically adjusting to changing threat landscapes, confirming robust and reliable cyberattack recognition and response mechanisms.
- The BMLSSA-CAD method innovates by incorporating COA-based feature selection with BBN for cyberattack recognition. This combination gives a structured model to optimize feature sets and model parameters dynamically, improving detection accuracy and efficiency in complex cybersecurity environments. By utilizing COA for dimensionality reduction and BBN for probabilistic reasoning, the model gives a robust framework to adaptively evaluate and respond to growing cyber threats, thereby improving cybersecurity abilities in real-time monitoring and threat mitigation.

Related works

Saheed et al.[11] proposed the IoT-defender architecture that integrates a Modified Genetic Algorithm (MGA) method with the LSTM model for determining cyber threats in IoT. The GA fitness function (FF) was employed for fine-tuning. To resolve the problem of class imbalance, the model used the focal loss operation that offers superior weights to minority categories, thereby enhancing the capability of the system to learn from the specific classes. Alani and Awad[12] considered IoT security and introduced an intelligent 2-layer IDS for IoT. The model's intelligence was obtained by ML methods for IDS, with a 2-layer model dealing with packet- and flow-based features. The individuality and originality of the technique were developed by integrating the ML and selection units for flow- and packet-based features. Golchha et al.[13] projected a cyberattack detection model for Industrial-

IoTs (IIoTs) using the Voting-based Ensemble Learning algorithm. An ensemble of the standard ML methods, including Random Forest (RF), Histogram Gradient Boosting (HGB), CatBoost, and hard voting methods, have been implemented to identify cyberattacks effectively. Feng et al.[14] designed an innovative adversarial security setup and developed a security game system which combines defence resource allocation and patrol assessment. SDSA computes the distribution approach of the best patrolling scheme that must be more appropriate for the protector by examining the strategy under the discrete action space and allows defence agents to proficiently cooperate via training the Dueling Double Deep Q-Network (D3QN). Awotunde et al.[15] developed ensemble methods that assisted with the FS model for IDS in the IIoT environment. The Chi-Square Statistical technique could be deployed for FS, and diverse ensemble methods like AdaBoost, Bagging, RF, eXtreme gradient boosting (XGBoost), and extra trees (ET)techniques must be executed for the identification of intrusion utilized to the datasets.

Ren et al.[16] introduced a multiagent deep reinforcement learning (DRL) automated security management technique. It forms a limited random game network attack-defense system. Leveraging RL methods, an autonomous defence agent will be developed. Besides, a network attack agent was designed. Additionally, drawing motivation in MINIMAX Q-learning, a cooperative training method, was considered for addressing the complexity of surrounding variability. Alattas and Mardani[17] presented an innovative system model dependent upon a stochastic estimation of finding the parameters reliant on a new adaptive-DL (ADL) method. An innovative architecture can be developed to integrate the component of arbitrariness rather than determined models. The proposed technique deliberated the network forensic systems and IDS. The method was introduced based on the five ordered protection phases. Xu et al.[18] projected a data-driven method for anomaly and intrusion detection in which various techniques could deal with and filter the information. The superiority of the training dataset was increased by employing the Synthetic Minority Oversampling Technique (SMOTE) model and mutual information (MI). Automatic ML has also been used for identifying the method with auto-tuned hyperparameters to be better satisfied for categorizing the data. In[19], the open set recognition (OSR) challenge in IoT-specific Network IDS (NIDS) is addressed by utilizing image-based data representations for extracting geographical traffic patterns. The Recurrent Neural Networks (RNNs) exhibited suboptimal accuracy and lacked parallelizability for attack evaluation tasks. The study also presents the Sparrow Search Optimization Algorithm (SSOA) as a basis for developing an effective assault classification technique.

Harahsheh, Al-Naimat, and Chen[20] propose an improved feature selection technique to mitigate the computational overhead on IoT resources while concurrently strengthening intrusion recognition abilities within the IoT environment. In[21], the Weighted Variational Autoencoder-based Hunter Prey Search (WVA-HPS) model is introduced. This method implements a weighted variational autoencoder (VA) with weight regularization and ensemble averaging, improved by the Hunter Prey Search optimization (HPSO) model to reduce overfitting and improve effectualness. Mohammed et al.[22] presented a Chaotic Sparrow Search Algorithm with DL utilizing the Recurrent Neural Network (RNN-CSSA) model. The technique uses the Binary Pigeon optimization Algorithm (BPEO) method for feature selection and the RNN model for classification. Arulkumar et al.[23] propose a model utilizing the Lanner Swarm Optimization (LSO) technique to optimize resource allocation and workload distribution. The LSO technique enhances effectualness. The objective function prioritizes diverse virtual machines (VMs) depending on their accomplishment times. Saheed, Omole, and Sabit[24] introduce the GA-mADAM-IIoT model for IIoT, by integrating a genetic algorithm for feature selection with a modified Adam optimizer for LSTM networks. The model also incorporates an attention mechanism to improve significant data processing and utilizes SHAP for improved transparency. Gaber et al.[25] propose a novel intrusion detection model based on the Particle Swarm Optimization (PSO) and Bat algorithm (BA) for feature selection and the RF classifier for the classification of malicious behaviours in IIoT-based network traffic. Altunay and Albayrak[26] developed three techniques for intrusion detection in IIoT networks by employing DL techniques, namely Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a hybrid CNN + LSTM.

Wankhade et al.[27] present an ML approach to detect attacks and anomalies by incorporating feature extraction, data preprocessing, and model training. Altunay et al.[28] utilized DL models such as CNN, Autoencoders (AE), Deep Belief Networks (DBN), and LSTM, which are used for extracting features from SCADA data for normal and abnormal classification. The classification process also implements techniques, namely the softmax function, Extreme Learning Machine (ELM), and Multilayer Perceptron (MLP). Qaddoori and Ali[29] introduce a security paradigm for edge devices against Message Queue Telemetry Transport (MQTT)-based attacks using an Intrusion Detection and Prevention System (IDPS). A methodology for training ML methods is also proposed on high-performance platforms. Furthermore, various security techniques confirm the authenticity and privacy of exchanged models and data. Altunay, Albayrak, and Çakmak[30] propose an AE-based IDS system to detect security anomalies in critical infrastructures. Ellappan et al.[31] introduce the sliding principal component and dynamic reward reinforcement learning (SPC–DRRL) methodology, which involves preprocessing using min-max normalization and a robust log-likelihood sliding principal component feature extraction algorithm. Finally, a dynamic reward reinforcement learning model is proposed. Alani, Mauri, and Damiani[32] present a two-stage system for detecting and classifying cyber-attacks based on ML. Khadidos et al.[33] introduce the binary hunter–prey optimization with a machine learning-based phishing attack detection (BHPO-MLPAD) method by utilizing a binary HPO model for feature selection. A cascaded forward neural network (CFNN) model is used for classification, with the variable step fruit fly optimization (VFFO) method used to adjust CFNN parameters.

Khan et al.[34] introduce a novel hybrid Trust Management Scheme to enhance trustworthiness and reliability (MASTER) in industrial sensor networks. By utilizing a clustering approach, MASTER efficiently detects and mitigates adversarial attacks. It also features a flexible weighting scheme that prioritizes recent interactions in direct and indirect trust evaluations. Alwasel et al.[35] employed a comprehensive experimental method. The model

also integrated graph features into ML techniques. Zhang et al.[36] introduce a CNN-based intrusion detection model with a Sparse Transformer (CST) that extracts local features via CNN and temporal features utilizing sparse self-attention. To address class imbalance in the dataset, the EQL v2 loss function is used to enhance the weights of minority classes. Pundir et al.[37] present MADP-IIME, a malware detection mechanism for IoT-enabled industrial multimedia environments that employ four ML models, such as Naive Bayes (NB), logistic regression (LR), artificial neural networks (ANN), and RF, to efficiently detect malware attacks. Ghasemkhani et al.[38] introduce Federated Multi-Label Learning (FMLL), a novel methodology integrating federated learning (FL) principles with a multi-label learning technique. Employing ML strategies, FMLL attained crucial enhancements in classification accuracy. Alrowais et al.[39] propose the MFO-RELM model, which integrates Mayfly optimization (MFO) with a regularized extreme learning machine (RELM) for cybersecurity threat detection in IoT environments. Tiwari et al.[40] developed a high-accuracy intrusion detection model. This method utilizes PSO for feature selection and employs feature reduction models such as PCA, LDA, and t-SNE. Moreover, the Generalized Additive Model (GAM) and Multivariate Adaptive Regression Splines (MARS) are utilized to detect potentially disruptive payloads. Table 1 summarizes the existing studies on cyberattack detection.

The existing studies incorporate an MGA with an LSTM methodology to improve cyber threat detection while addressing class imbalance utilizing focal loss. A two-layer intelligent IDS utilizes the ML technique for packet and flow-based feature analysis, although scalability could be a concern. An ensemble method implementing diverse standard ML approaches aims for effectual cyberattack detection but may complicate model interpretation. Other studies present adversarial safety setups incorporating resource allocation and patrol strategies, while some employ feature selection methods that could overlook relevant attributes, resulting in computational overhead. Multiagent reinforcement learning for security management exhibit's ability but may face difficulty with real-time implementation due to complexity. Techniques, namely stochastic estimation and data-driven methods, encounter threats in variability and overfitting, while hybrid models integrating several optimization approaches risk inconsistent detection rates. The dependence on complex architectures and ensemble methods can hinder effectiveness and interpretability in dynamic IoT environments. Despite enhancements in IoT security, there still needs to be a substantial gap in addressing the real-time adaptability and interpretability of IDSs. Many existing models depend on complex approaches that, while efficient in controlled environments, need help performing consistently in dynamic, real-world scenarios. Moreover, the incorporation of growing threats and the balance between detection accuracy and computational efficiency still need to be explored, underscoring the requirement for more robust and streamlined approaches in growing IoT landscapes.

The proposed model

This study presents a novel BMLSSA-CAD method for IIoT networks. The technique mainly intends to improve security in the IIoT platform by detecting cyberattacks. The BMLSSA-CAD technique contains procedures like min-max normalization, COA-based FS, BBN-based cyberattack detection, and SSA-based hyperparameter tuning. Figure 1 illustrates the workflow of the BMLSSA-CAD method.

Min-max normalization

Initially, the BMLSSA-CAD technique undergoes a min-max scalar that can be used to normalize the input data. Min-max scaling (feature scaling or min-max normalization) is a commonly used data preprocessing in statistics and ML models[41]. Min-max scaling aims to convert the numerical value of a feature into a particular range, usually between 0 and 1. This can be done by subtracting the least values from the data points and dividing the results by the range (the difference between the minimum and maximum values). Min-max scaling is especially suitable when handling features with varying scales, as it ensures that each feature equally contributes to the analysis. This normalization method helps enhance the performance of ML approaches, particularly those sensitive to the measure of input features and can improve convergence during training.

Feature selection using COA

The BMLSSA-CAD technique involves a COA-based FS method to elect an optimal feature subset. Chameleon is a hierarchical clustering method that uses a qualitative model[42]. This model is advantageous for feature selection because it can effectively balance exploration and exploitation. Unlike conventional models, COA adapts dynamically to the landscape of the feature space, allowing it to detect optimal feature subsets that enhance model performance while minimizing redundancy. Its hybrid nature incorporates the merits of swarm intelligence and local search strategies, making it robust against local minima. Furthermore, COA is appropriate for high-dimensional datasets, often in complex domains like IoT security. By concentrating on the most informative features, COA can improve computational efficiency and mitigate model complexity, ultimately improving generalization and accuracy. These merits make COA a compelling choice over other feature selection techniques, particularly in scenarios where data quality and relevance are critical. In this work, if the computations and intersections between them correspond to the computations and intersections of cluster items, then the two clusters are merged. Then, the data item is converted into a small sub-cluster from the shared images, and later, the sub- class is combined with the hierarchical clustering to get the actual outcome. The unified model assists in detecting homogeneous or natural groups and is employed for all types of data if the feature is similar. The Chameleon algorithm considers cluster computation and connectivity, particularly the inherent properties of clusters, to detect related sub-clusters. Figure 2 indicates the workflow of the COA approach.

The Chameleon method describes the property as a k-nearest neighbour graph. The K- point signifies a data object in the nearest neighbour graph, and if the data A is the k- nearest object of data B, then the A and B objects are the edges. The closest image of the K-community can be attained dynamically. Community: The concept of K- concept is formulated: the local electricity of an object can be defined by the place density where the object is. The sibling density defines the electrical community of objects. In dense population areas, the

Ref. No.	Objective	Method	Dataset	Measures
11	To develop the IoT-defender framework for detecting cyberattacks in IoT networks	MGA, LSTM, fine-tuning of LSTM parameters using a GA fitness function	BoT-IoT, UNSW-NB15, N-BaIoT	Accuracy, detection rate, precision score, and false alarm rate
12	To develop an intelligent two-layer IDS for IoT	Two-layer architecture, ML, feature selection	Standard dataset	Packet and flow-based accuracy
13	To develop a cyber-attack detection framework for the IIoT	HGB, CatBoost, and RF, hard voting classifier	CICIDS2017 dataset	Accuracy, recall, precision, F1-score, AUC, ROC, and MCC
14	To develop a DRL-based security defence strategy	SDSA, D3QN	IoT traffic datasets	Standard measures
15	To develop a model for efficient intrusion detection in IIoT networks	Chi-square statistical method, XGBoost, bagging, ET, RF, AdaBoost	TON_IoT dataset	Accuracy, recall, precision, and F1-score
16	To develop a multiagent DRL approach for autonomous security management in decentralized networks	Finite random game network attack–defence model, synchronized interactive training mechanism	Experimental simulations in various network configurations	Standard measures
17	To develop a stochastic framework utilizing a novel ADL model	Network forensic systems	Bot-IoT dataset	Simulation of IoT network traffic
18	To develop a data-driven approach	SMOTE, automated ML	IoT network traffic datasets	Standard measures
19	To address the open set recognition challenge in IoT-specific NIDS	SSOA, RNN	Benchmark dataset	Classification accuracy
20	To enhance intrusion detection capabilities in IoT environments	Supervised classification techniques	InSDN dataset	Accuracy
21	To strengthen cybersecurity threat detection in IoT environments	WVA, HPSO	BoT-IoT, MQTTset, IoT-23	Precision, accuracy, specificity, F-measure, and recall
22	To accurately detect anomalies in IoT-enabled smart cities	CSSA, RNN, BPEO	UCI-SECOM, UNSW NB-15	Accuracy
23	To enhance resource allocation and workload distribution in cloud-assisted CIoT	LSO, load balancing and workload scheduling techniques	Benchmark dataset	Makespan, response time, resource utilization rate, execution time, latency, throughput, and delivery rate
24	To propose a GA-mADAM-IIoT for effective intrusion threat detection in IIoT networks	GA, LSTM, mADAM, CCE, SHAP	SWaT, WADI	Accuracy, AUC, recall, precision, F1-score, MCC
25	To propose a novel intrusion detection model	PSO, BA, RF	WUSTL-IIOT-2021 dataset	Accuracy, recall, precision, and F1-score
26	To develop and evaluate three deep learning models for intrusion detection in IIoT networks	CNN, LSTM, CNN + LSTM	UNSW-NB15, X-IIoTID dataset	Accuracy
27	To enhance the security of IIoT networks	ML models	IIoT network data	Attack detection rates and False positive rates
28	To analyze the efficiency of various DL models for anomaly-based intrusion detection systems in SCADA networks	CNN, AE, DBN, LSTM, softmax function, ELM, MLP	SCADA network datasets	Positive and negative aspects of each approach
29	To develop a security paradigm for edge devices that utilizes ML models to detect MQTT-based attacks	ML, IDPS	MQTT attack datasets	Standard measures
30	To develop and evaluate an AE-based IDS	AE	UNSW-NB15 dataset	Accuracy
31	To develop and evaluate an SPC-DRRL model for enhancing the detection performance in IIoT networks	SPC, DRRL	TON_IoT dataset	Attack detection time, computational overhead, and error rate
32	To develop a two-stage ML system for the efficient detection and classification of cyber-attacks on smart grids	ML	DNP3 intrusion detection dataset	Detection and attack type classification score
33	To develop a BHPO-MLPAD method for detecting phishing attacks in IoT	BHPO, CFNN, VFFO	UNSW dataset	Accuracy, precision, recall, F-score, and AUC
34	To enhance security, trustworthiness, and collaboration in IWSNs through a novel hybrid Trust Management Scheme	Multi-layered assessment and clustering approach, ML models	Varying percentages of malicious sensor nodes	Malicious behavior detection rate, FNR, throughput rates, and energy consumption
35	To enhance the accuracy of portscan attack detection in IIoT networks	Graph representation, data preprocessing, ML techniques	ISOT-CID	Standard measures
36	To develop an effective intrusion detection model for IoT networks	CNN, sparse transformer, EQL v2 loss function	Edge_IIoT, UNSW-NB15, CICIDS-2017, CICIDS-2018	Detection accuracy, recall rate and F1 score
37	To develop a robust MADP-IIME for IoT-enabled industrial multimedia environments	NB, LR, ANN, RF	Standard dataset	Accuracy, precision, recall, and F1 score
38	To develop and evaluate an FMLL approach	FL principles, multi-label classification strategy, base classifier	Amphibians, Anuran-Calls-(MFCCs), HackerEarth-Adopt-A-Buddy datasets	Accuracy, precision, recall, and F-score
39	To develop and evaluate the MFO-RELM model for effective cybersecurity threat detection and classification in IoT	Preprocessing of IoT data, RELM, MFO	N-BaIoT dataset	Accuracy, precision, recall, and F-score
40	To develop a high-accuracy intrusion detection technique for IIoT networks	PSO, PCA, LDA, t-SNE, GAM, MARS	WUSTL-IIOT-2021	Accuracy, latency reduction

Table 1. Summary of existing studies on cyberattack detection in IIoT.

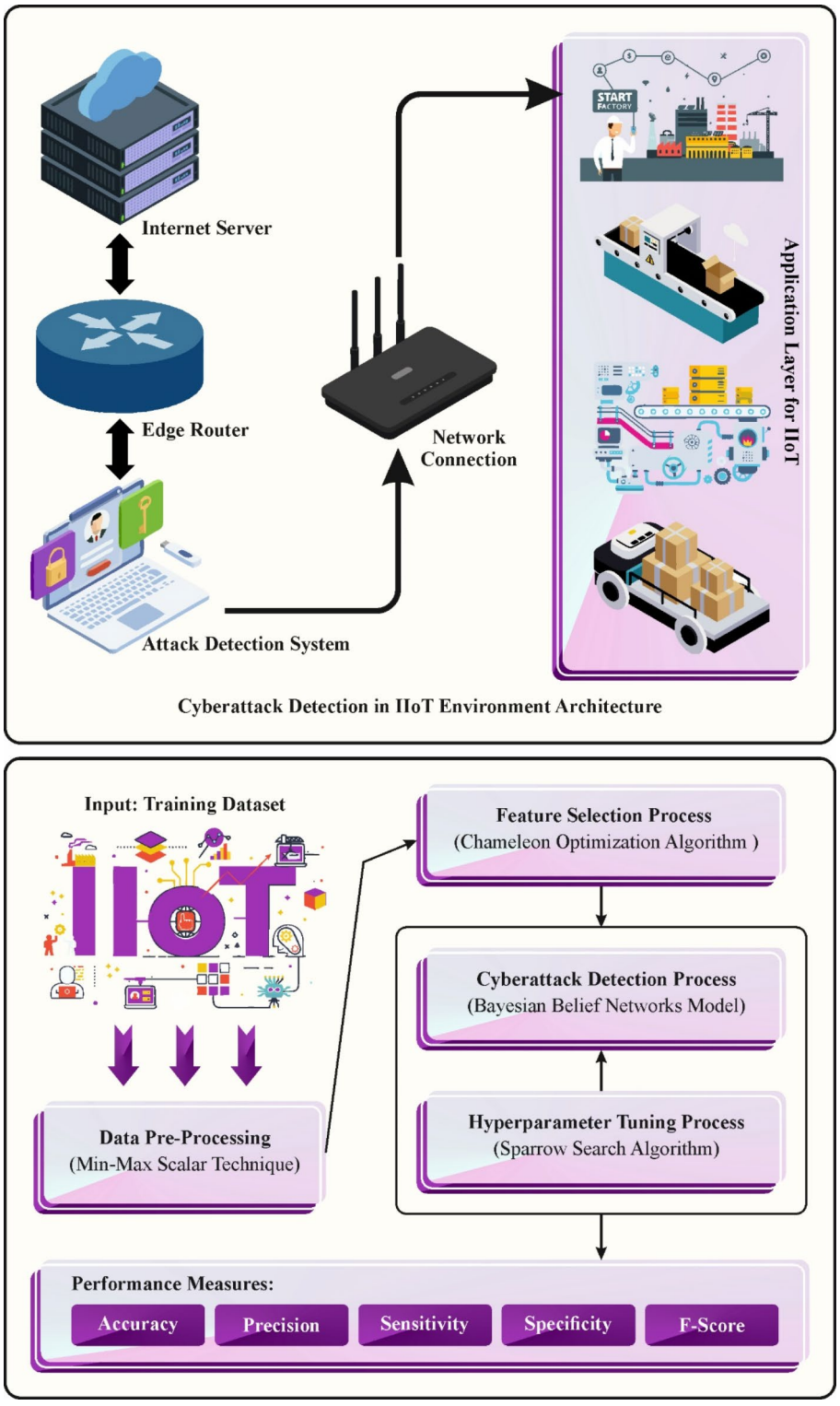

Fig. 1. Working flow of BMLSSA-CAD technique.

community can be defined narrowly. In the object distribution, the group defined is more comprehensive, and the density area is represented by edge weight. The Chameleon defines the similarity among the clusters using the relative approximation $RC(C_i, C_j)$ and the relative connection $RI(C_i, C_j)$ of both clusters.

(1) Relative interconnection $RI(C_i, C_j)$ defines the standardization of the internal connection of both clusters and the absolute connection between C_i and C_j

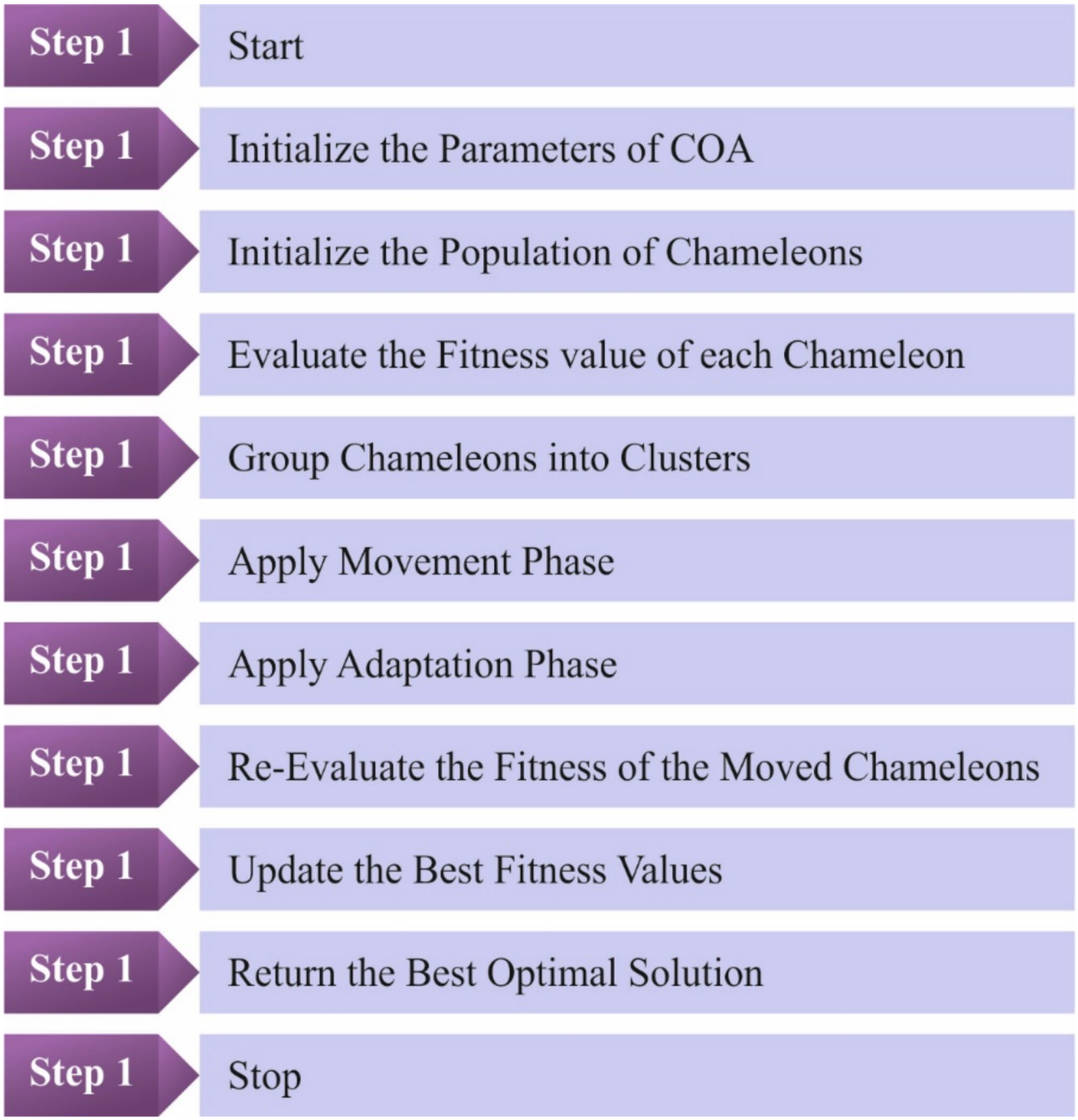

Fig. 2. Workflow of COA technique.

$$RI\left(C_i,\, C_j\right) = \frac{\left|E_{C_i,C_i}\right|}{\frac{1}{2}\left|E_{C_i}\right| + \left|E_{C_j}\right|} \tag{1}$$

Where EC_{C_i,C_i} denotes the truncated edge of the cluster having C_i and C_j categorized into C_i and C_j;EC_{C_i}(or EC_{C_j}) shows the size of minimal truncated bisector (the weight amount of edges that should approximately split into two equivalent parts)

(2) Relative approximation (C_i,C_j) defines the Normalization of absolute approximation between C_i and C_j regarding the internal approximation of both clusters.

$$RC\left(C_i,\, C_j\right) = \frac{S_{EC(C_1,C_j)}}{\frac{|C_i|}{|C_i|+|C_j|}S_{EC} + \frac{|C_j|}{|C_i|+|C_j|}S_{EC_{C_j}}} \tag{2}$$

Where the average weight of edges interconnecting vertices and minimal truncated bisector C_i and $C_j$are indicated asS_{EC} and $S_{EC(C_1,C_j)}$,correspondingly.
The FF assumes the classifier results and the number of selected attributes. It increases the classifier efficiency and reduces the size of chosen attributes. Then, the following FF is used to assess the solutions.

$$Fitness = \alpha * ErrorRate + (1-\alpha) * \frac{\#SF}{\#All\_F} \tag{3}$$

In Eq. (3), $ErrorRate$ indicates the classifier error value using the chosen attributes and is calculated as the ratio of incorrect classifier to the number of classifications made, ranging within [0,1]. α controls the impact of classifier quality and subset length, and α is fixed at 0.9. $\#All\_F$ is the overall quantity of features from the new data, and $\#SF$ refers to the number of attributes chosen.

Cyberattack detection using BBN model

The BMLSSA-CAD technique uses the BBN model for cyberattack detection[43]. This approach is an ideal choice for cyberattack detection due to its capacity to handle uncertainty and incomplete data, which are general in real-world scenarios. BBNs utilize a probabilistic framework that allows for integrating prior knowledge and updating beliefs based on new evidence, making them adaptable to growing threats. Figure 3 illustrates the structure of BBM model. This characteristic enables the approach to efficiently analyze complex relationships between diverse variables, such as attack vectors and system vulnerabilities. Furthermore, BBNs provide clear interpretability, allowing security analysts to comprehend the rationale behind detection decisions. Compared to conventional ML techniques, BBNs can present enhanced robustness in dynamic environments where data may fluctuate. This makes them specifically suited for cybersecurity applications where precise risk assessment is significant. As a directed acyclic graph (DAG), BBN contains a collection of nodes and conditional probability, indicating joint distribution probability amongst the node variables. The parent node and child node are two kinds of nodes in BBN. One of the significant aspects of BBN is that joint distribution probability is easily determined. In BBN, joint distribution probability $P\left(X\right)$,$X = (X_1, X_2, X_3, \ldots, X_n)$ when the probability of X_i parent node is described by $Pa\left(X_i\right)$:

$$P\left(X\right) = (X_1, X_2, X_3, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i | Pa\left(X_i\right)\right) \tag{4}$$

In Eq. (4), $X = (X_1, X_2, X_3, \ldots, X_n)$ represents the BBN variable, and the amount of variables in BBN is n . When there is new evidence, then the probability can be dynamically updated. If the event Y is given to BBN, then $P\left(X|\,Y\right)$ of event X is represented as:

$$P\left(X|Y\right) = \frac{P\left(X\right)P\left(Y|X\right)}{P\left(Y\right)} = \frac{P\left(X\right)P\left(Y|X\right)}{\sum_{i=1}^{n} P\left(Y|X_i\right)} \tag{5}$$

In Eq. (5), $P\left(Y\right)$ and $P\left(X\right)$ marginal probability and the previous probability of events Y and X.

The architecture of BBN mainly consists of two stages: (1) Parameter learning defines the conditional probability at node variable. (2) Structure learning defines the factor nodes (variables) and finds the independent or dependent relationships between them to design a DAG. The construction of BBN has the following:

1. The variable node of BBN is defined by expert experience and domain knowledge (DK) or prior knowledge.
2. The BBN is obtained by automatically learning the sample dataset through ML methods.
3. The structure of BBN is acquired through the data fusion method using ML and DK.

Meanwhile, the third technique incorporates the strengths of DK and ML; it removes the pitfalls that arise by using specific processes. The popular ML methods, including hill-climbing and $K2$, perform structural learning from the dataset. The $K2$ model performs structured learning that searches according to the nodes' order through a limited number of parent nodes. The $K2$ model exploits posterior probability as a scoring function:

(1) Compute the Cooper-Herskovits (CH) score for X_j based on the order of node ρ.

$$CH = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left[log\frac{\Gamma\left(\alpha_{ij*}\right)}{\Gamma\left(\alpha_{ij*}+m_{ij*}\right)} + \sum_{k=1}^{r_i} log\frac{\Gamma\left(\alpha_{ijk}+m_{ijk}\right)}{\Gamma\left(\alpha_{ijk}\right)}\right] \tag{6}$$

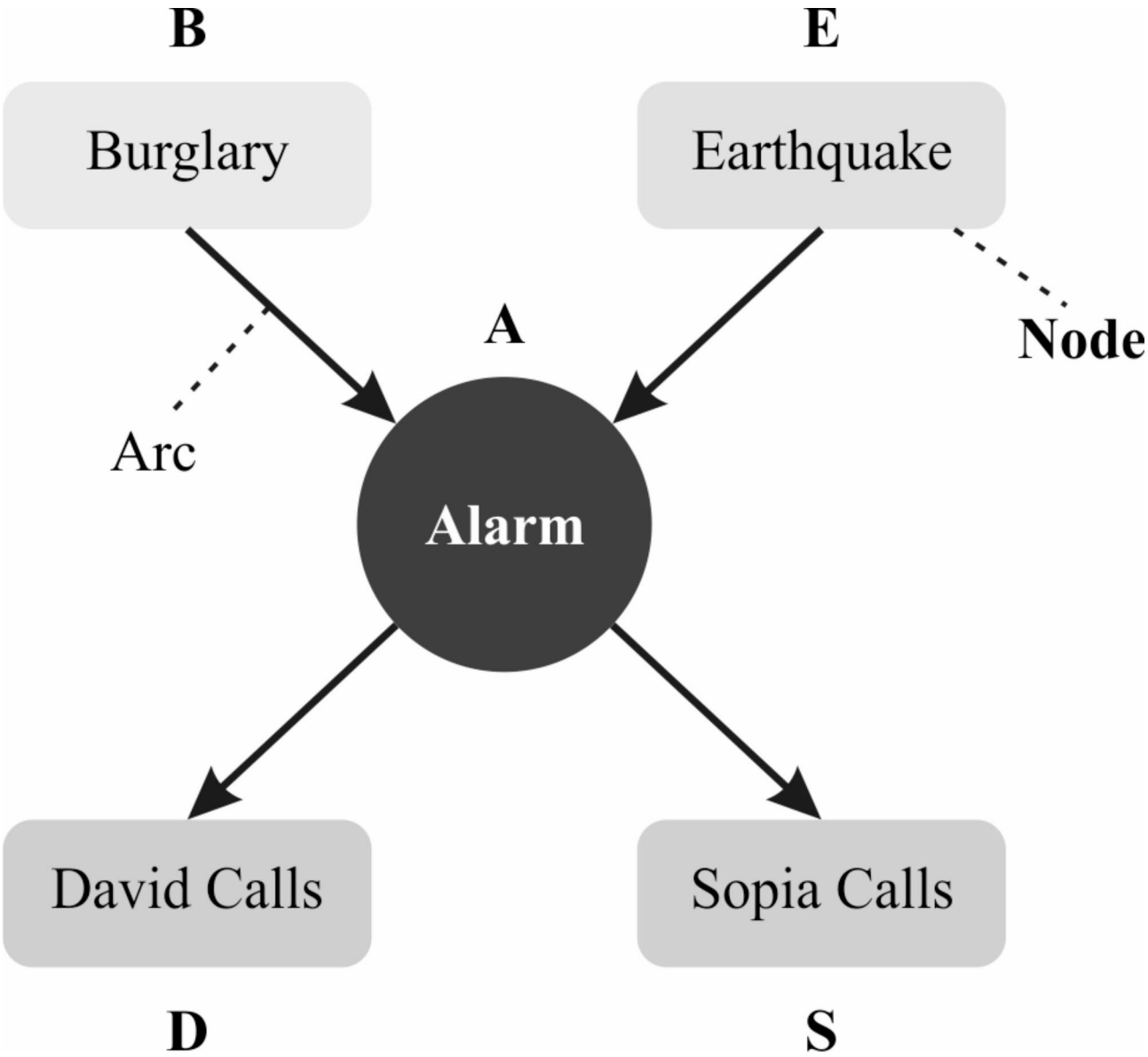

Fig. 3. Architecture of BBN technique.

In Eq. (6), the amount of samples m_{ijk} subjected to $X_i = k$,$\pi\ (X_i) = j$,$m_{ij*} = \sum_{k=1}^{r_i} m_{ijk}$,$\alpha_{ij*} = \sum_{k=1}^{r_i} \alpha_{ijk}$, and $\alpha_{ijk} = P(X_i = k|\pi\ (X_i) = j)$.

(2) If $X_i (i \neq j)$, then add arc $(X_i \rightarrow X_j)$, which makes the $CH\ (X_j, \pi_j \cup X_i)$maximum. π_j is the parent of X_j.

Hyperparameter tuning process

Finally, the hyperparameter tuning process is performed by utilizing the SSA model to enhance the performance of the BBN technique[44]. This utilization presents various merits over conventional optimization techniques. SSA is inspired by the foraging behaviour of sparrows, allowing it to effectively explore the solution space and avoid local optima, thereby enhancing the convergence rate. Its population-based approach improves exploration and exploitation, resulting in a more robust search for optimal hyperparameters. Unlike gradient-based techniques, SSA does not need derivative data, making it appropriate for intrinsic and non-differentiable objective functions often faced in BBNs. Moreover, SSA effectively handles high-dimensional spaces, which is significant for BBNs with various parameters. Overall, its adaptability and effectiveness make SSA an ideal option for improving BBN performance compared to other hyperparameter tuning methods.

The SSA simulates the behaviour of sparrow populations that are separated into scroungers and producers when discovering food. Mainly, the producers are highly liable for locating food in an extensive atmosphere. They want to find the sparrow population's position and way of food. Every separate sparrow's portion is not stable, and they want to be capable of adapting to change as per the condition. Naturally, sparrows on the border are

very weak to attack; they slowly alter their locations to change nearer to the midpoint of the populace to upsurge their safety. Furthermore, every sparrow knows that if a single sparrow identifies risk, the entire population travels from its place to a secure situation to endure searching.

SSA progresses by initially setting a cluster of randomly generated particles and series the highest iteration count. N denotes the population dimension. Every particle has speed and location assets.

Producer: They have a higher foraging exploration region when compared to the scrounger because to meet its food requirements, it also wants to deliver the way of the foraging area for the whole populace. The location of the i^{th} particles is upgraded at every iteration as:

$$x_i^{k+1} = \begin{cases} x_i^k \cdot exp\left(\frac{-i}{\alpha \cdot k_{\max}}\right) & if \quad R < ST \\ x_i^k + Q \cdot L & if \quad R > ST \end{cases} \tag{7}$$

Here, k denotes the existing iterations count, $k_{\max}$specifies the highest number of iterations, α refers to the randomly produced value from the interval of zero and one, Q represents the randomly generated number focused on the usual distribution, L refers to the matrix of 1×D where entire elements are 1, R refers to the alarm value in the interval of [0and1], and ST states to be secure threshold within [0.5, 1].

If $R < ST$, the region has no risk, and the producer searches nearby. If $R \geq ST$, the producer intellects hazard and changes to an arbitrary path.

Scrounger: All particles without producers are said to be scroungers. They always follow the producer data. When the scrounger observes that the producers have originated a superior foraging region, it offers an existing location and travels to the superior foraging region to stare for food. The location of the ith scrounger particles (x_i) has been upgraded throughout every iteration as follows:

$$x_i^{k+1} = \begin{cases} Q \cdot exp\left(\frac{p_w^k - x_i^k}{i^2}\right) & if \quad i > \frac{N}{2} \\ p_b^{k+1} + \left|x_i^k - p_b^{k+1}\right| \cdot A' \cdot L & if \quad i \leq \frac{N}{2} \end{cases} \tag{8}$$

Here, p_b refers to the global finest place, p_w denotes the worst location, and A' denotes the $D \times D$ matrix with a random number1 or -1.

The volume of food attained by the scrounger is too small ($i > \frac{N}{2}$), then it flies towards another location to discover food. If $i \leq \frac{N}{2}$, the scrounger monitors the producer to an optimum foraging region.

Watchman: In the set instructions, every particle has an investigation and initial cautionary device. It may be alert of hazards and so unrestraint the existing region and travel to a secure area. A particle is named watchman. The location of the ith watchman particle is upgraded as below:

$$x_i^{k+1} = \begin{cases} p_b^k + \beta \cdot \left|x_i^k - p_b^k\right| & if \quad f_i > f_b \\ p_b^k + K \cdot \frac{x_i^k - p_w^k}{|f_i - f_w| + \epsilon} & if \quad f_i = f_b \end{cases} \tag{9}$$

Here, β denotes the factor of step size regulation, and its significance is a randomly generated integer with a normal distribution by the variance of 1 and the mean of 0. K refers to the randomly produced number within [1 and 1]. At the same time, f_w and f_b correspondingly specify the worst and best fitness values. A small constant ϵ is used to stop the denominator from being 0.

If $f_i > f_b$, the particle is situated at the border of the populace, and it especially travels nearer to the midpoint. When $f_i = f_b$, the particle is located in the centre of the populace, and it travels arbitrarily to acquire near to other particles to evade being hunted.

The comprehensive calculation workflow of SSA is given below. Figure 4 depicts the flowchart of SSA.

Step 1: Set the populace by initializing the size N, the highest iteration count, a protection threshold, and the ratio of producers and sparrows alert of hazards.

Step 2: Compute the fitness of the existing populace's individual and type to discover the present worst and best values.

Step 3: Pick the particle with decent fitness value as a producer as per the percentage and upgrade the location as per Eq. (7);

Step 4: Give the residual particles as scroungers and upgrade their locations as per Eq. (8);

Step 5: Arbitrarily pick a few individuals as particles that are alert of hazards per the percentage and give them as watchmen. Upgrade their locations based on Eq. (9) and compute the novel fitness value. Upgrade the locations when the fitness is superior to the present optimum value.

Step 6: Compute the value of fitness and preserve the location of the optimum individuals;

Step 7: Confirm that the termination condition is stratified, then stop the process and return to the optimum outcome. Otherwise, go to Step 2.

The SSA develops an FF to accomplish enriched classifier accuracy and describes a positive integer to characterize the higher performance of the solution candidate. Now, the reduction of classifier error is taken as a FF, as follows:

$$\begin{aligned} fitness\,(x_i) &= ClassifierErrorRate\,(x_i) \\ &= \frac{No.\,of\,misclassified\,samples}{Total\,No.\,of\,samples} \times 100 \end{aligned} \tag{10}$$

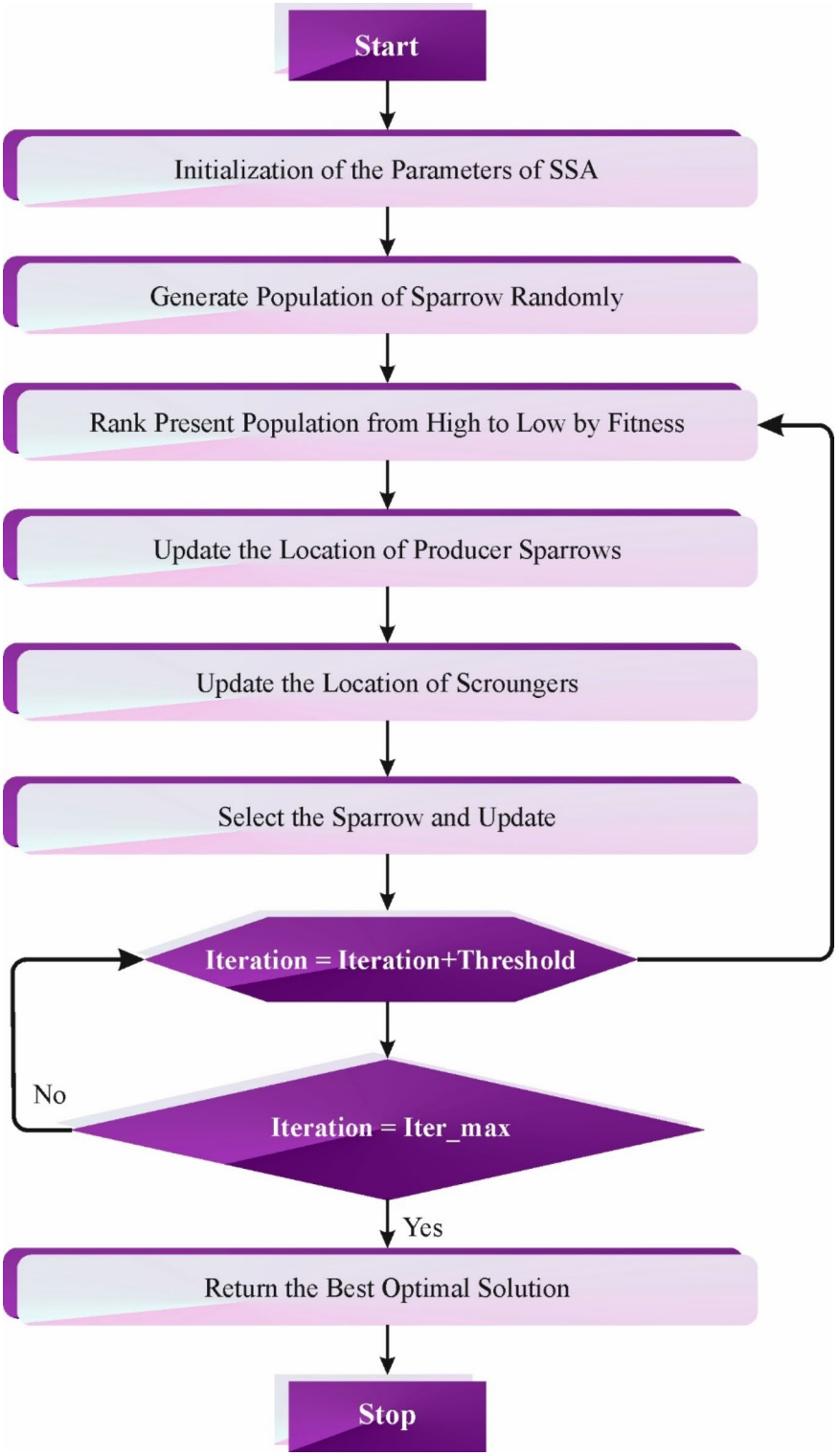

Fig. 4. Flowchart of SSA.

Performance validation

The performance validation of the BMLSSA-CAD method utilizes dual benchmark datasets such as the UCI SECOMD and UNSWNB51 datasets[45].

Dataset description

The UNSWNB51 dataset contains 10,000 samples under ten classes, as defined in Table 2. The UNSW-NB15 dataset contains 42 features (excluding labels) across 10 class labels, encompassing nine attack types and one normal category, such as Normal, Fuzzers, Analysis, Exploits, Backdoors, Generic, Shellcode, DoS, Worms, and Reconnaissance. Additionally, the UCI-SECOM dataset consists of 591 features with two classes, providing a rich resource for evaluating ML techniques in the context of intrusion detection and anomaly classification.

The datasets are chosen due to their relevance in cybersecurity, especially for network intrusion detection. Recognized for its representation of real-world network traffic patterns and diverse cyber threats, this dataset is ideal for training and assessing models focused on efficiently recognizing anomalies and attacks. Its extensive coverage of several attack scenarios and network activities confirms thorough testing of the performance and applicability of the BMLSSA-CAD model in complex cybersecurity environments.

Data Analysis

Figure 5 establishes the classifier results of the BMLSSA-CAD model below the UNSWNB51 dataset. Figure 5a and b portrays the confusion matrices offered by the BMLSSA-CAD model on 70%ofTRAS:30%ofTESS. The experimental value indicated that the BMLSSA-CAD method has detected and classified each of the ten classes. Also, Fig. 5c and d demonstrates the attack recognition analysis of the BMLSSA-CAD model on 70:30 of TRAS/TESS. The figure stated that the BMLSSA-CAD approach has detected ten classes proficiently.

The attack detection outcomes of the BMLSSA-CAD technique on the UNSWNB51 dataset are described in Table 3; Fig. 6. The simulation value implies that the BMLSSA-CAD technique recognizes ten classes proficiently. With 70%TRAS, the BMLSSA-CAD technique gains an average $accu_y$ of 99.56%, $prec_n$ of 97.84%, $sens_y$ of 97.82%, $spec_y$ of 99.76%, and F_{score} of 97.81%. Also, with 30%TESS, the BMLSSA-CAD method obtains an average $accu_y$ of 99.55%, $prec_n$ of 97.80%, $sens_y$ of 97.76%, $spec_y$ of 99.75%, and F_{score} of 97.77%.

The classifier outcomes of the BMLSSA-CAD technique are graphically offered in Fig. 7 in the training accuracy (TRAAC) and validation accuracy (VALAC) curves on the UNSWNB51 dataset. The figure displays a clear understanding of the behaviour of the BMLSSA-CAD method over various epochs, representing its learning procedure and generalization abilities. The figure especially concludes a constant advancement in the TRAAC and VALAC with increasing epochs. It shows the diverse nature of the BMLSSA-CAD method in the pattern detection procedure on both datasets. The increase in VALAC summarizes the capability of the BMLSSA-CAD model to adjust to the TRA dataset. It also precisely classifies hidden datasets, showing strong generalization skills.

Figure 8 exhibits the training loss (TRLOS) and validation loss (VALOS) outcomes of the BMLSSA-CAD method over different epochs on the UNSWNB51 dataset. The steady decrease in TRLOS shows that the BMLSSA-CAD method improved the weights and lessened the classifier error on both datasets. The figure interprets the BMLSSA-CAD model's relationship with the TRA dataset, emphasizing its capability to take patterns within both datasets. The BMLSSA-CAD approach repetitively increases its parameters to decrease the differences between the forecast and actual TRA classes.

Inspecting the PR curve, as depicted in Fig. 9, the outcomes certified that the BMLSSA-CAD approach gradually achieves improved PR values below every class on the UNSWNB51 dataset. It demonstrates the better capabilities of the BMLSSA-CAD method in classifying different classes, exhibiting the capability to distinguish classes.

In addition, in Fig. 10, ROC curves formed by the BMLSSA-CAD technique outperformed the identification of dissimilar labels on the UNSWNB51 dataset. This delivers a comprehensive understanding of TPR and FRP tradeoffs over discrete detection thresholds and epochs. The figure emphasizes the boosted performance of the BMLSSA-CAD method below all classes, delineating its efficacy in addressing many classification problems.

UNSWNB15 dataset	
Classes	**No. of samples**
Normal	1000
Generic	1000
Exploits	1000
Fuzzers	1000
DoS	1000
Reconnaissance	1000
Analysis	1000
Backdoor	1000
Shellcode	1000
Worms	1000
Total samples	10,000

Table 2. Details on the UNSWNB51 dataset.

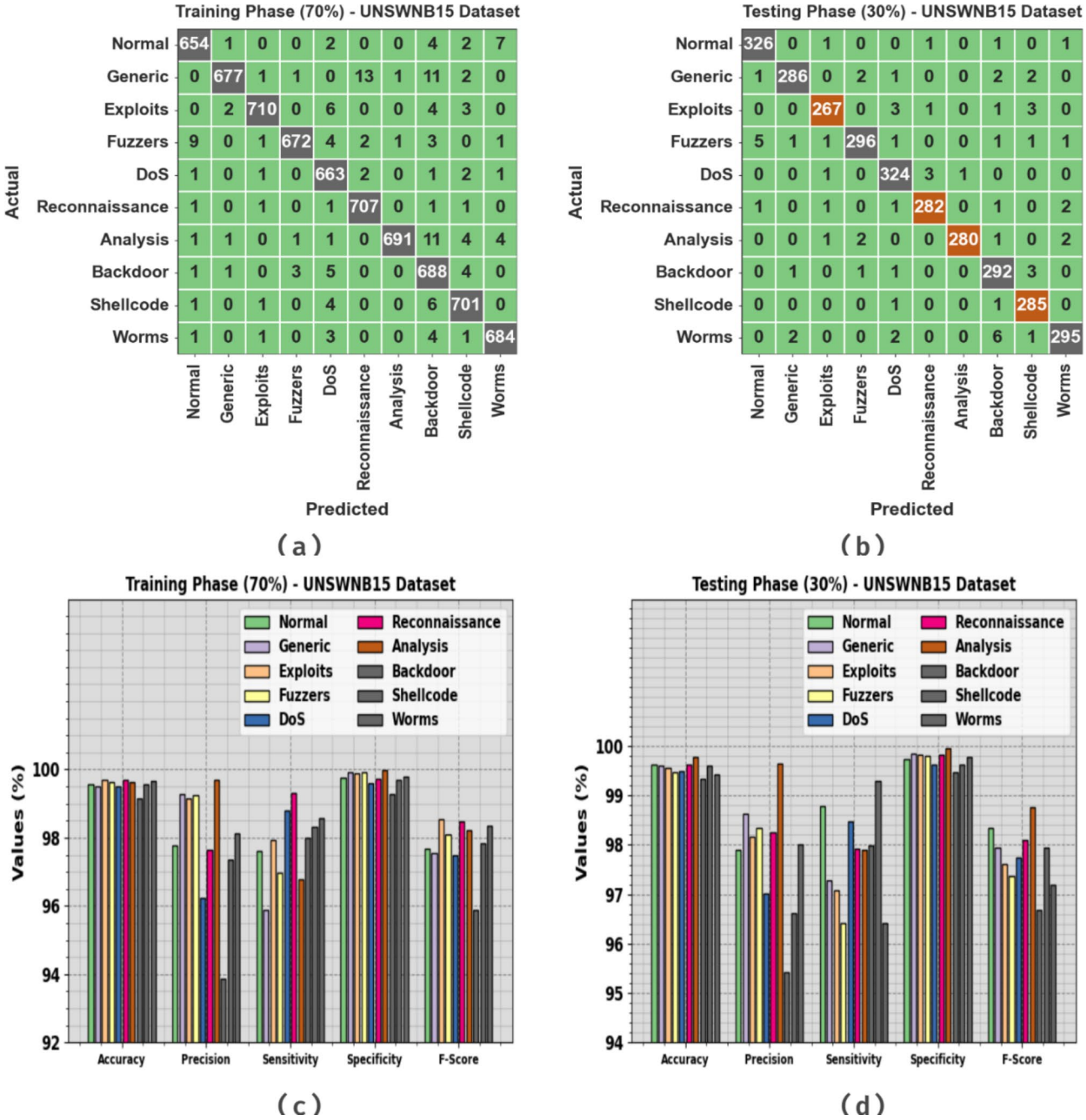

Fig. 5. UNSWNB51 dataset (**a**, **b**) Confusion matrices and (**c**, **d**) Classifier outcome.

Table 4; Fig. 11 show a detailed review of the BMLSSA-CAD method with existing methods on the UNSWNB51 dataset[45]. The experimental value stated that the BMLSSA-CAD method reaches enhanced performance. It is noticed that the ANN and KNN models have shown reduced performance. Simultaneously, DT, VLSTM, SSA-CRNN, MFSDL-ADIIoT, and GJODL-CADC models have achieved considerable performance. But, the BMLSSA-CAD approach surpassed the other models with maximum $prec_n$, $reca_l$, $accu_y$, and F_{score} of 97.84%, 97.82%, 99.56%, and 97.813%, correspondingly.

The UCI SECOM dataset comprises 5000 samples under two classes, as expressed in Table 5[45].

Figure 12 exhibits the performance of the BMLSSA-CAD approach below the UCI SECOM dataset. Figure 12a and b exemplifies the confusion matrices the BMLSSA-CAD approach provides on 70%of TRAS:30%of TESS. The simulation outcome implied that the BMLSSA-CAD method has precisely recognized and classified all 2-class labels. Similarly, Fig. 12c and d exhibits the attack recognition analysis of the BMLSSA-CAD methodology at 70:30 of TRAS/TESS. The figure stated that the BMLSSA-CAD methodology identified two classes proficiently.

The attack recognition outcomes of the BMLSSA-CAD methodology on the UCI SECOM dataset are described in Table 6; Fig. 13. The outcomes imply that the BMLSSA-CAD approach identifies dual classes proficiently.

UNSWNB15 Dataset					
Classes	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{Score}
TRAS (70%)					
Normal	99.56	97.76	97.61	99.76	97.68
Generic	99.51	99.27	95.89	99.92	97.55
Exploits	99.70	99.16	97.93	99.90	98.54
Fuzzers	99.63	99.26	96.97	99.92	98.10
DoS	99.51	96.23	98.81	99.59	97.50
Reconnaissance	99.69	97.65	99.30	99.73	98.47
Analysis	99.64	99.71	96.78	99.97	98.22
Backdoor	99.16	93.86	98.01	99.29	95.89
Shellcode	99.56	97.36	98.32	99.70	97.84
Worms	99.67	98.13	98.56	99.79	98.35
Average	**99.56**	**97.84**	**97.82**	**99.76**	**97.81**
TESS (30%)					
Normal	99.63	97.90	98.79	99.74	98.34
Generic	99.60	98.62	97.28	99.85	97.95
Exploits	99.57	98.16	97.09	99.82	97.62
Fuzzers	99.47	98.34	96.42	99.81	97.37
DoS	99.50	97.01	98.48	99.63	97.74
Reconnaissance	99.63	98.26	97.92	99.82	98.09
Analysis	99.77	99.64	97.90	99.96	98.77
Backdoor	99.33	95.42	97.99	99.48	96.69
Shellcode	99.60	96.61	99.30	99.63	97.94
Worms	99.43	98.01	96.41	99.78	97.20
Average	**99.55**	**97.80**	**97.76**	**99.75**	**97.77**

Table 3. Attack detection outcome of BMLSSA-CAD technique on the UNSWNB51 dataset. Significant values are in bold.

With 70%TRAS, the BMLSSA-CAD approach obtains an average $accu_y$ of 98.97%, $prec_n$ of 98.97%, $sens_y$ of 98.97%, $spec_y$ of 98.97%, and F_{score} of 98.97%. Also, with 30%TESS, the BMLSSA-CAD method gains an average $accu_y$ of 98.93%, $prec_n$ of 98.93%, $sens_y$ of 98.93%, $spec_y$ of 98.93%, and F_{score} of 98.93%.

The performance of the BMLSSA-CAD technique is graphically shown in Fig. 14 in the TRAAC and VALAC curves method on the UCI SECOM dataset. The figure shows beneficial clarification into the behaviour of the BMLSSA-CAD technique over numerous epochs, validating its learning process and generalization skills. The figure determines a progressive enhancement in the TRAAC and VALAC with increasing epoch counts. It guarantees the adaptive nature of the BMLSSA-CAD method in the pattern detection procedure on both datasets. The increasing tendency in VALAC describes the capability of the BMLSSA-CAD method to adapt to the TRA dataset, which also excels in providing precise identification of hidden datasets, representing strong generalization abilities.

Figure 15 provides a detailed review of the TRLOS and VALOS outcomes of the BMLSSA-CAD technique over different epochs on the UCI SECOM dataset. The gradual decrease in TRLOS highlights the BMLSSA-CAD technique's improved weights and decreased classifier error on both datasets. The figure specifies an extensive knowledge of the BMLSSA-CAD model's relationship with the TRA dataset, underlining its ability to take patterns within both datasets. Notably, the BMLSSA-CAD methodology repeatedly improves its parameters in decreasing the alterations among the forecast and real TRA classes.

The results of inspecting the PR curve, as exposed in Fig. 16, showed that the BMLSSA-CAD method gradually achieves improved PR values below every class on the UCI SECOM dataset. This confirms the improved skills of the BMLSSA-CAD technique in classifying separate classes and demonstrates its ability to detect classes.

Besides, in Fig. 17, ROC curves formed by the BMLSSA-CAD methodology outperformed in identifying different labels on the UCI SECOM dataset. This provides extensive knowledge of the tradeoff between TPR and FRP over separate detection thresholds and epochs. The figure highlighted the improved performance of the BMLSSA-CAD technique below all classes, delineating its efficacy in addressing the classification problem.

Table 7; Fig. 18 show the comparative results of the BMLSSA-CAD method with current methods on the useful UCI SECOM dataset. The outcome concluded that the BMLSSA-CAD methodology attains greater performance. It is observed that the DNN and ensemble techniques have shown condensed performance. Simultaneously, PSO ensemble, SSA-CRNN, and GJODL-CADC methodologies have attained considerable performance. However, the BMLSSA-CAD approach exceeded the other models with the highest $prec_n$, $reca_l$, $accu_y$, and F_{score} of 98.93%, 98.93%, 98.93%, and 98.93%, respectively. Thus, the BMLSSA-CAD approach was executed for an enhanced detection process.

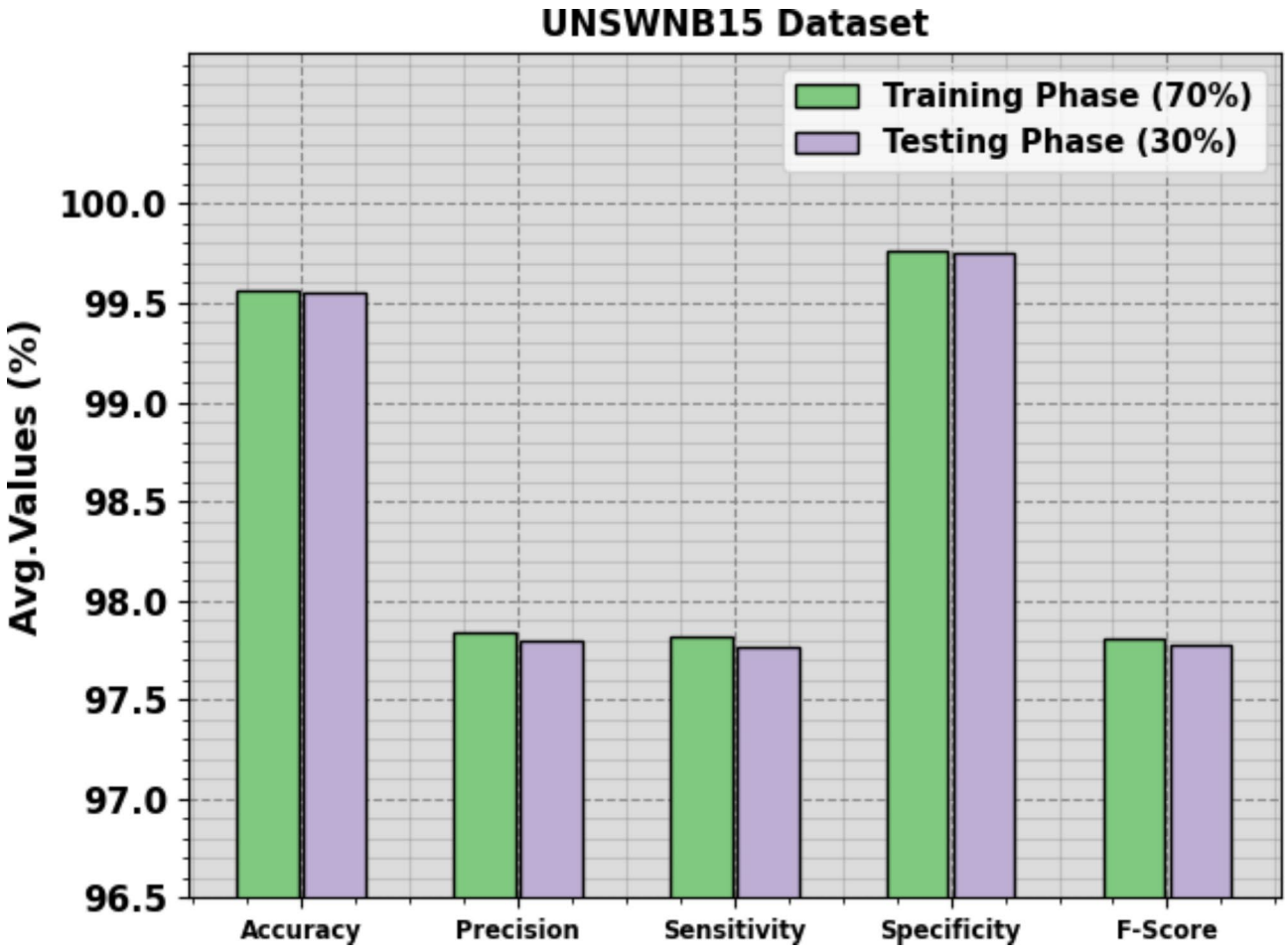

Fig. 6. Average of BMLSSA-CAD method on UNSWNB51 dataset.

Conclusion

This study presents a novel BMLSSA-CAD method in the IIoT environment. The presented BMLSSA-CAD method mainly intends to improve security in the IIoT platform by detecting cyberattacks. The BMLSSA-CAD technique contains procedures such as min-max normalization, COA-based FS, BBN-based cyberattack detection, and SSA-based hyperparameter tuning. Initially, the BMLSSA-CAD technique utilizes a min-max scalar to normalize the input data. Also, the BMLSSA-CAD technique employs a COA-based FS approach to elect an optimum feature subset. The BMLSSA-CAD technique uses the BBN model for cyberattack detection. The hyperparameter tuning method is performed by using the SSA to improve the performance of the BBN model. The performance of the BMLSSA-CAD method can be studied using a benchmark dataset. The experimental validation of the BMLSSA-CAD method highlighted superior accuracy outcomes of 97.84% and 98.93% compared to recent techniques on the IIoT platform. The limitations of the BMLSSA-CAD approach comprise potential threats in scaling to massive datasets due to the computational demands of the COA model for feature selection and the SSA for hyperparameter tuning. Furthermore, while BBN is effectual for modelling reliabilities among features, they may encounter limitations in comprehending convolutional associations in highly dynamic and growing cyberattack scenarios. Future studies may concentrate on optimizing the effectualness of COA and SSA methods, exploring ensemble models to improve the robustness of the model, incorporating real-time data streams for continuous monitoring, and addressing interpretability threats to enhance trust and usability in practical cybersecurity applications.

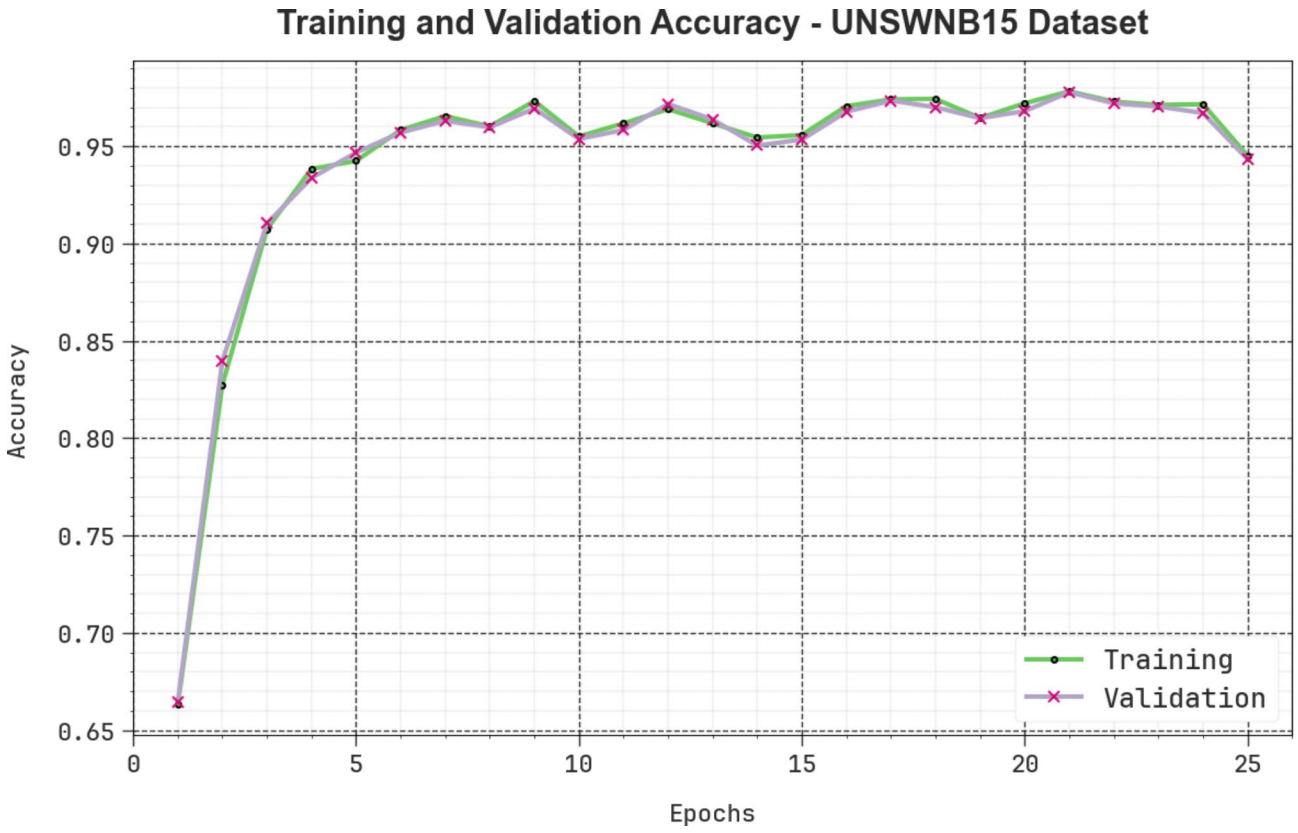

Fig. 7. $Accu_y$ curve of BMLSSA-CAD method on UNSWNB51 dataset

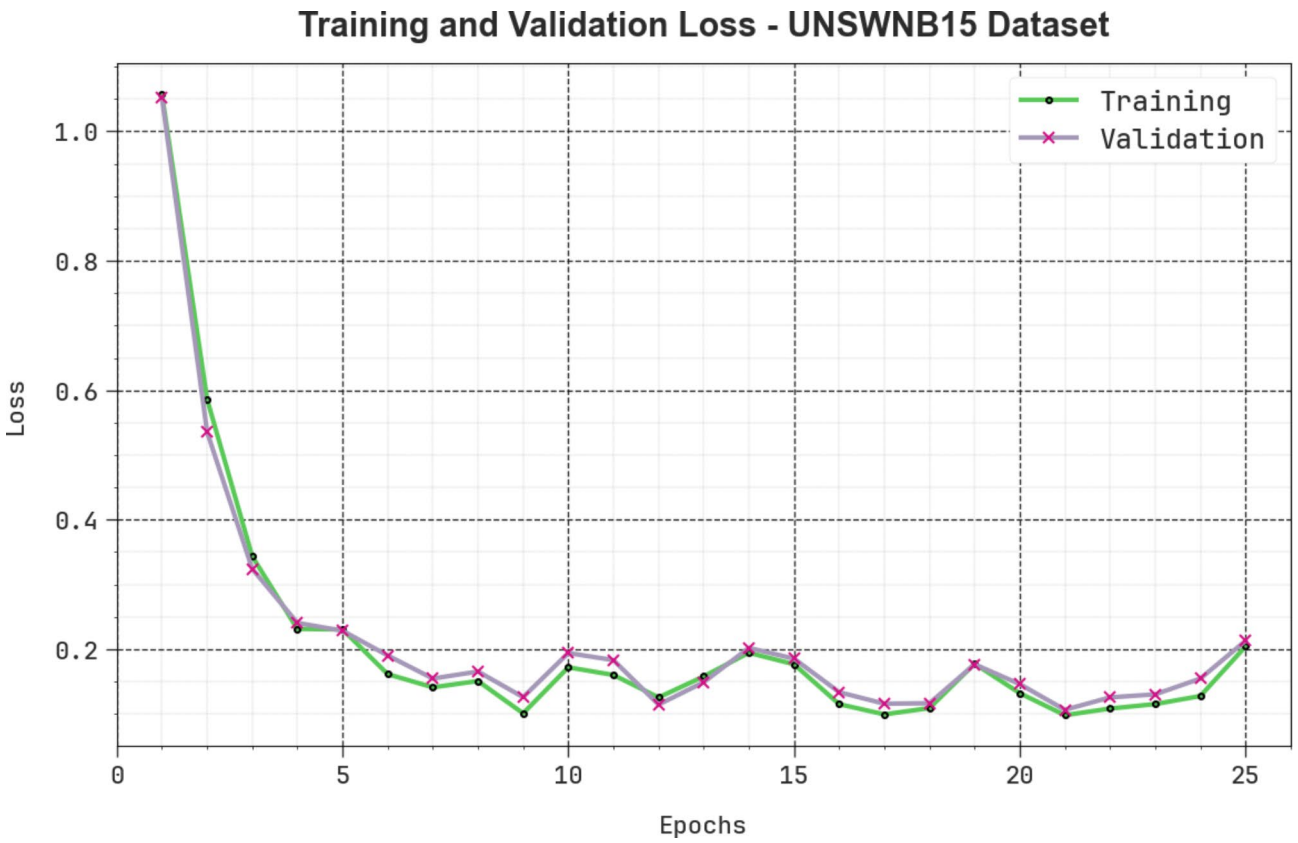

Fig. 8. Loss curve of BMLSSA-CAD technique on UNSWNB51 dataset.

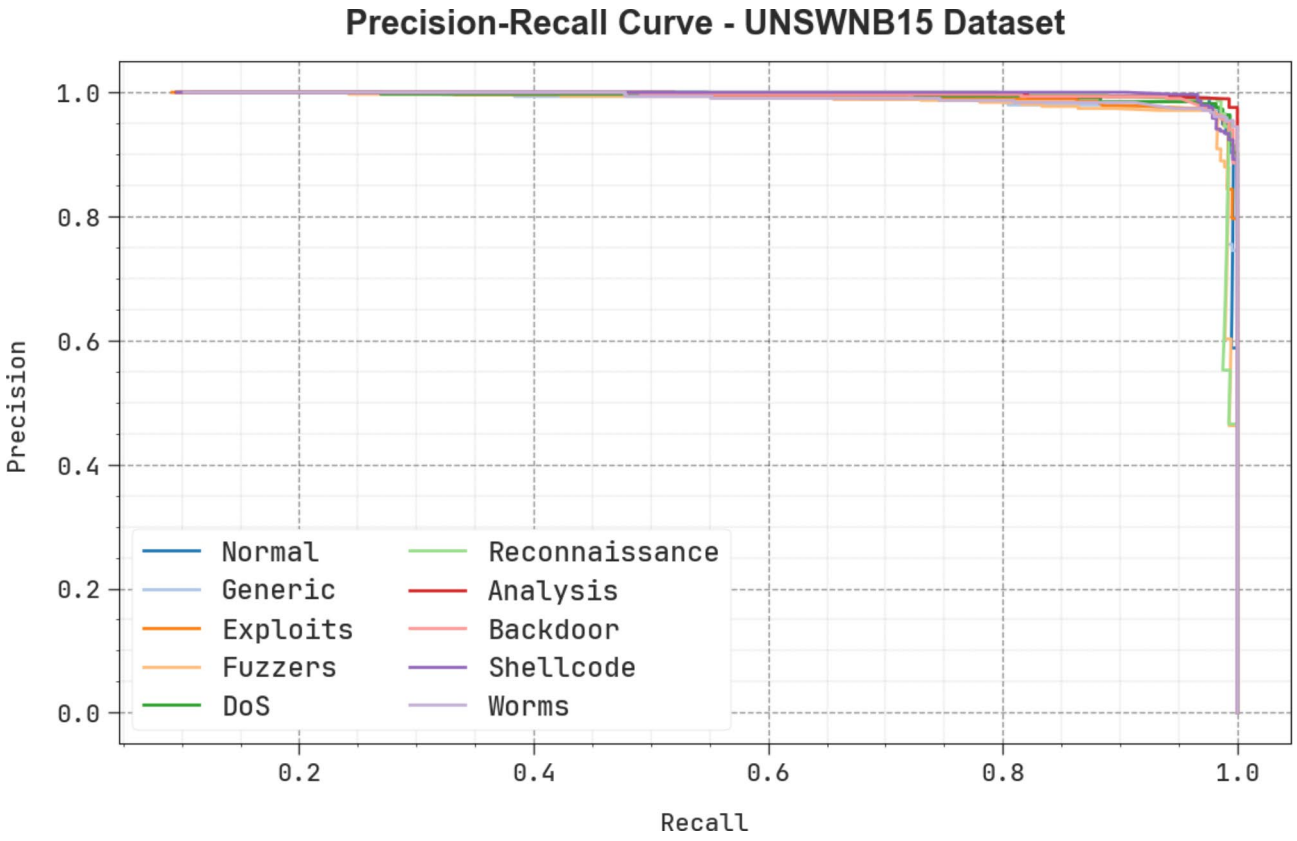

Fig. 9. PR curve of BMLSSA-CAD technique on UNSWNB51 dataset.

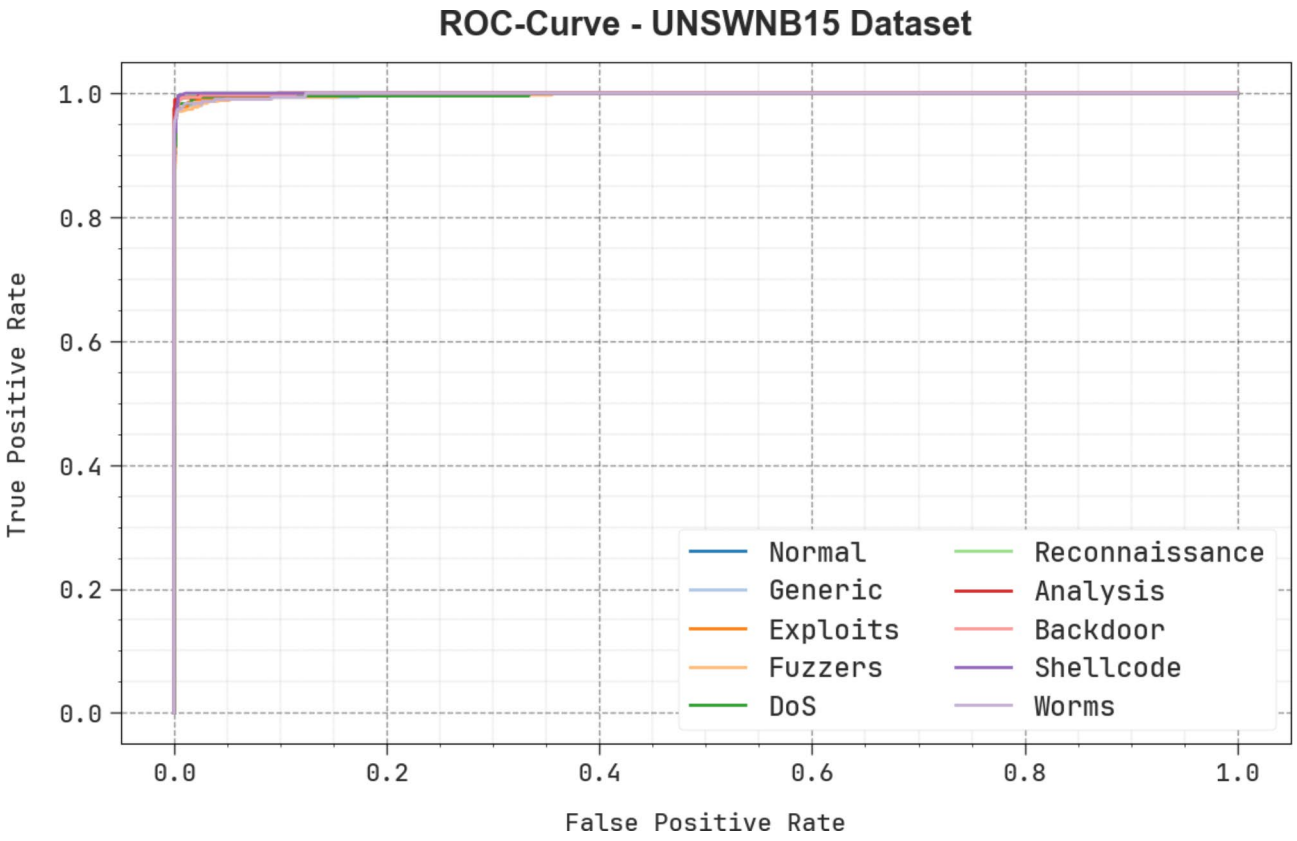

Fig. 10. ROC curve of BMLSSA-CAD technique on UNSWNB51 dataset.

UNSWNB15 Dataset				
Methods	$Prec_n$	$Reca_l$	$Accu_y$	F_{Score}
ANN	57.84	58.92	79.26	54.98
KNN	63.03	53.26	71.22	52.93
DT	64.29	53.57	70.74	48.83
VLSTM	67.08	53.27	96.08	58.83
SSA-CRNN	67.14	59.17	98.82	59.91
MFSDL-ADIIoT	67.11	60.35	99.10	60.37
GJODL-CADC	97.30	97.17	99.34	97.20
BMLSSA-CAD	97.84	97.82	99.56	97.81

Table 4. Comparative analysis of the BMLSSA-CAD model with existing approaches on the UNSWNB51 dataset[45].

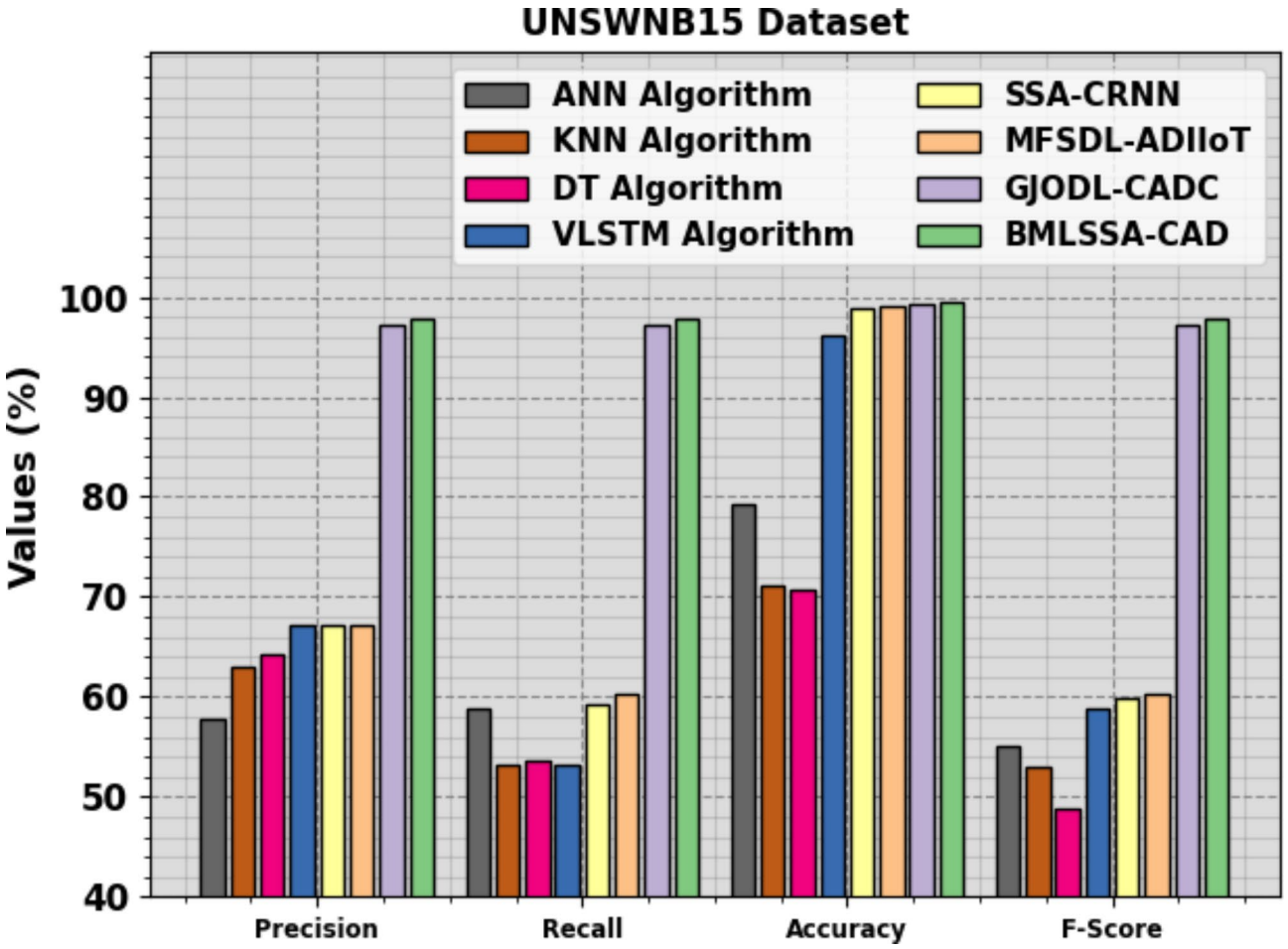

Fig. 11. Comparative analysis of the BMLSSA-CAD method on the UNSWNB51 dataset.

UCI SECOM Dataset	
Classes	**No. of samples**
Class 1	2500
Class 2	2500
Total samples	**5000**

Table 5. Details of the UCI SECOM dataset. Significant values are in bold.

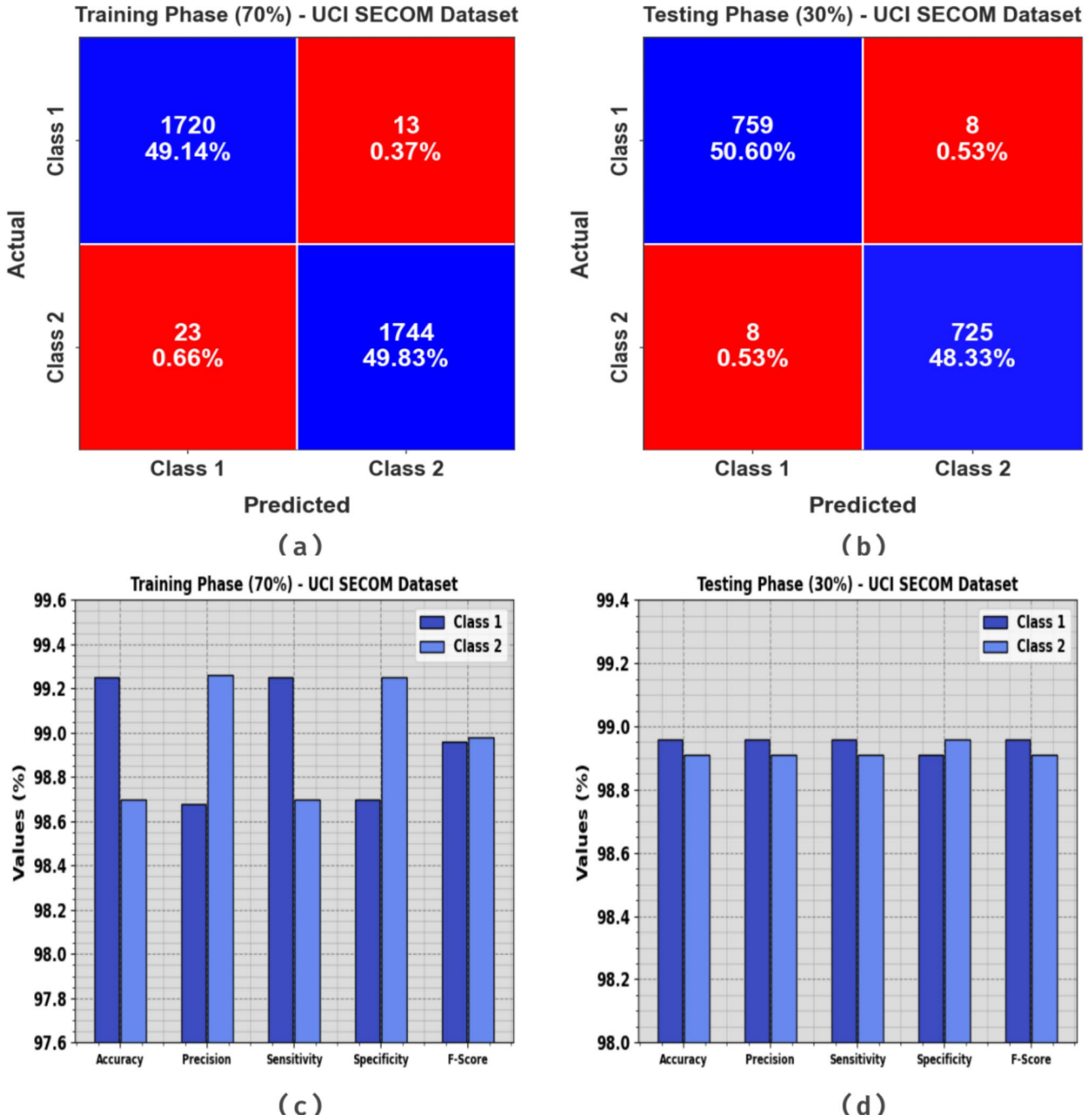

Fig. 12. UCI SECOM dataset (**a**, **b**) Confusion matrices and (**c**, **d**) Classifier outcome.

UCI SECOM Dataset					
Classes	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{Score}
TRAS (70%)					
Class 1	99.25	98.68	99.25	98.70	98.96
Class 2	98.70	99.26	98.70	99.25	98.98
Average	**98.97**	**98.97**	**98.97**	**98.97**	**98.97**
TESS (30%)					
Class 1	98.96	98.96	98.96	98.91	98.96
Class 2	98.91	98.91	98.91	98.96	98.91
Average	**98.93**	**98.93**	**98.93**	**98.93**	**98.93**

Table 6. Attack detection outcome of BMLSSA-CAD technique on the UCI SECOM dataset. Significant values are in bold.

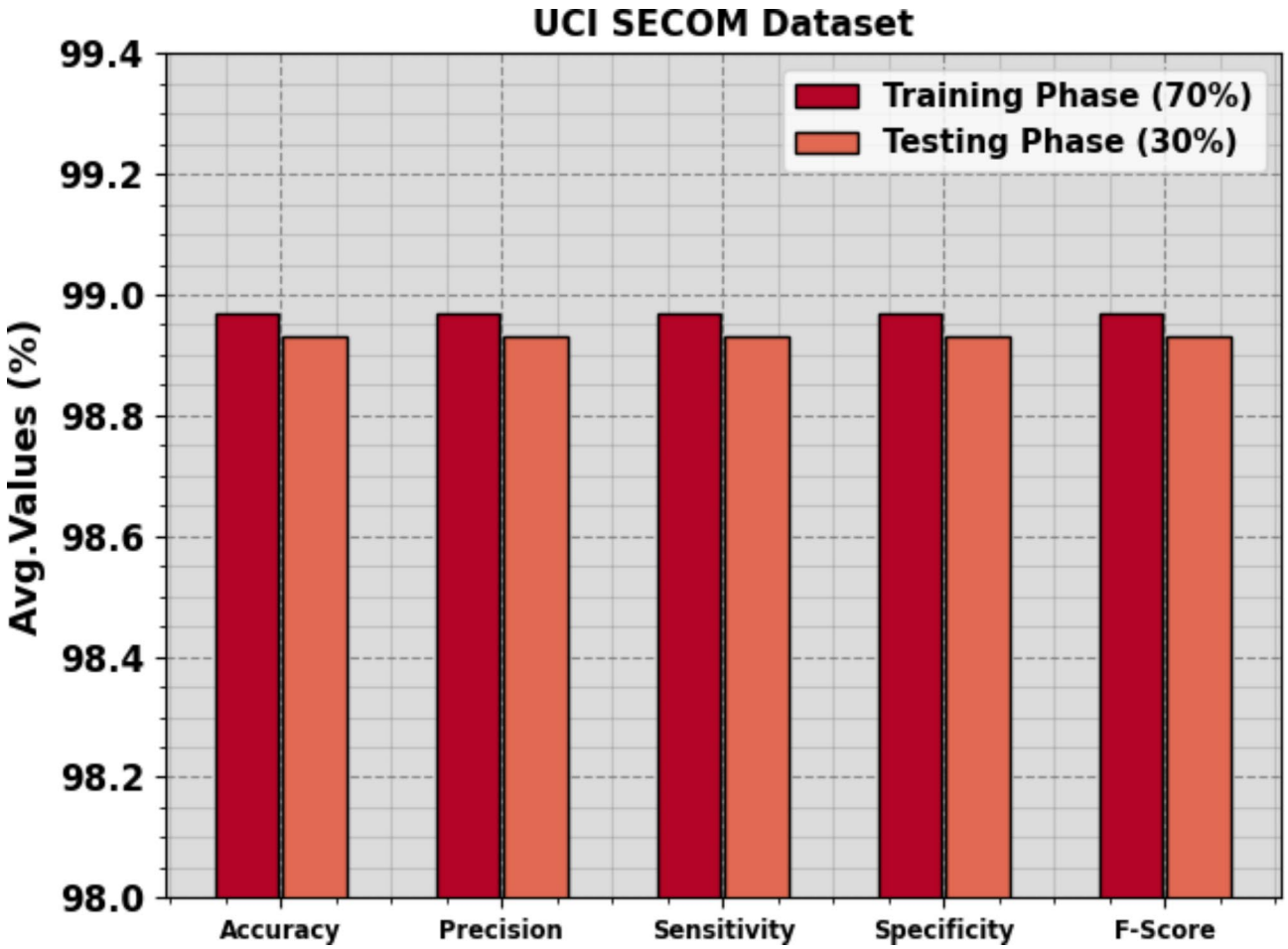

Fig. 13. Average of BMLSSA-CAD technique on UCI SECOM dataset.

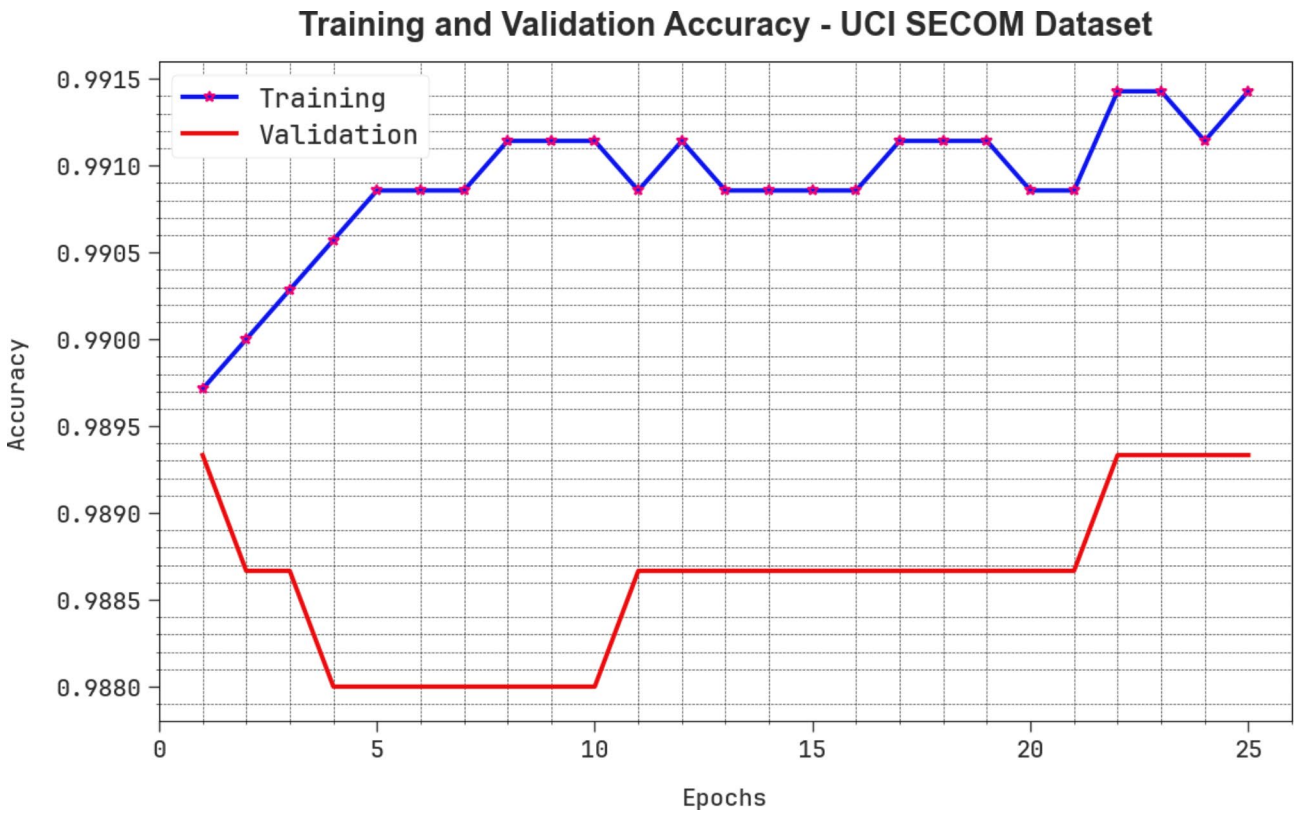

Fig. 14. $Accu_y$ curve of BMLSSA-CAD technique on UCI SECOM dataset

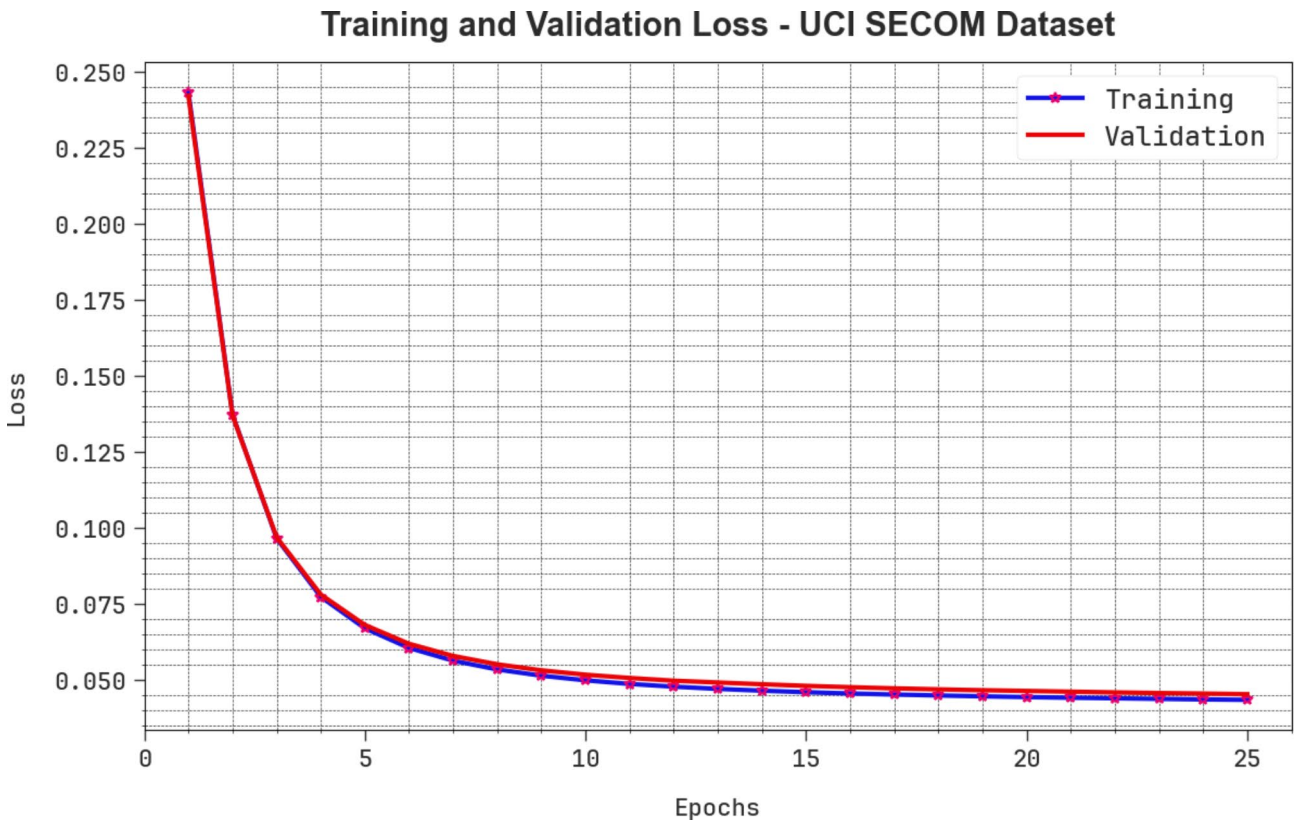

Fig. 15. Loss curve of BMLSSA-CAD technique on UCI SECOM dataset.

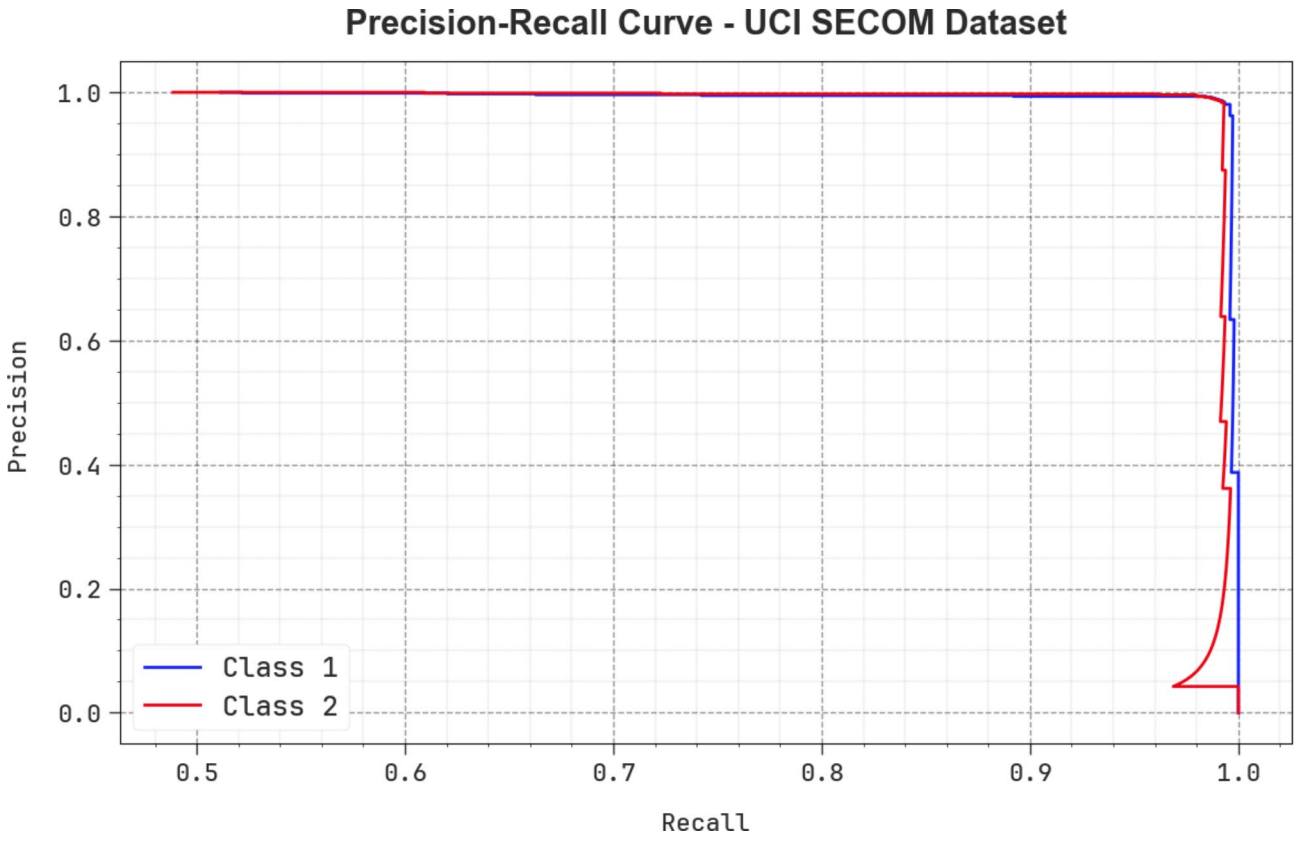

Fig. 16. PR curve of BMLSSA-CAD technique on UCI SECOM dataset.

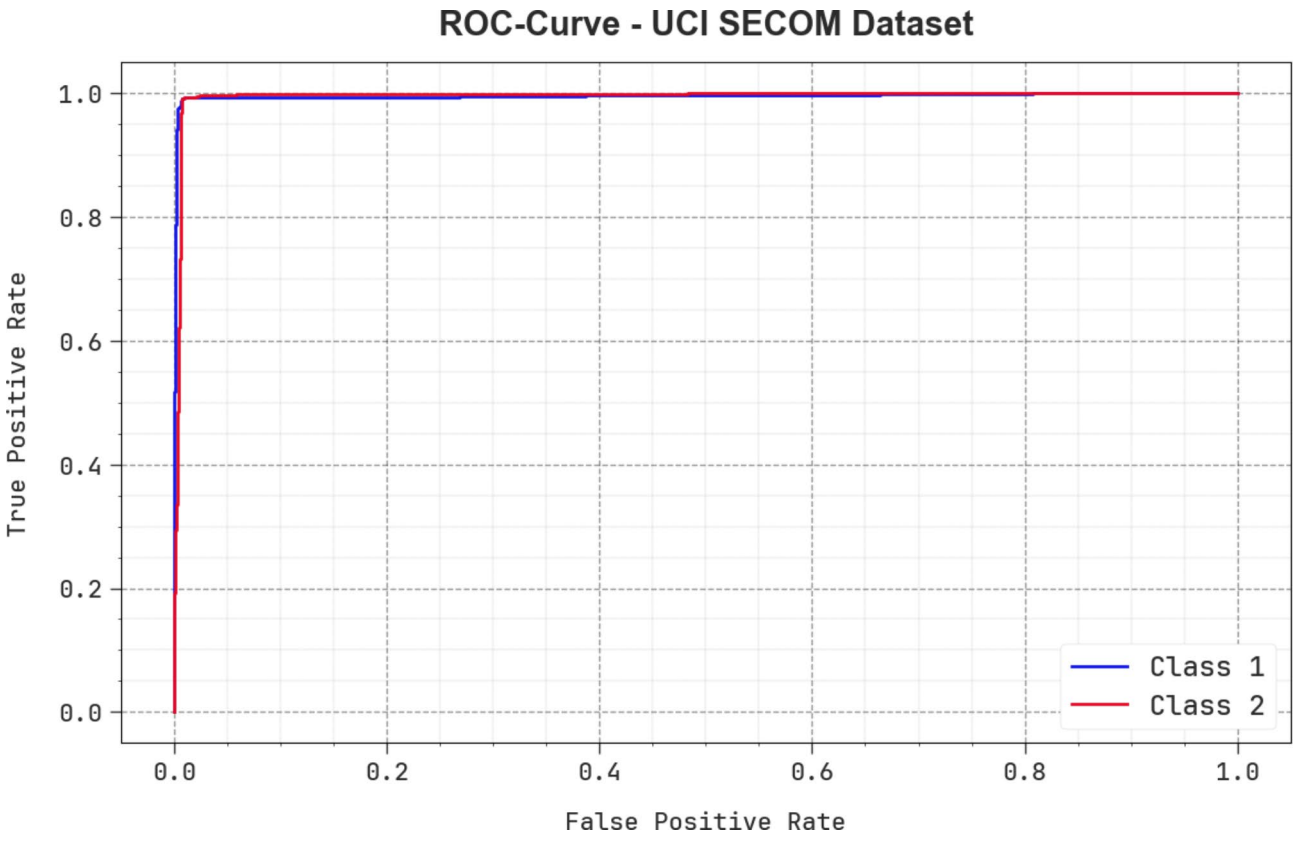

Fig. 17. ROC curve of BMLSSA-CAD technique on UCI SECOM dataset.

Methods	$Prec_n$	$Reca_l$	$Accu_y$	F_{Score}
DNN Layer	90.37	83.66	92.23	89.85
Ensemble	90.14	88.08	91.14	90.61
PSO Ensemble	91.11	86.84	93.48	90.69
SSA-CRNN	91.79	89.14	97.07	91.00
MFSDL-ADIIoT	92.44	89.89	97.90	91.12
GJODL-CADC	98.43	98.42	98.54	98.42
BMLSSA-CAD	98.93	98.93	98.93	98.93

Table 7. Comparative analysis of the BMLSSA-CAD approach with existing methods on the UCI SECOM dataset[45].

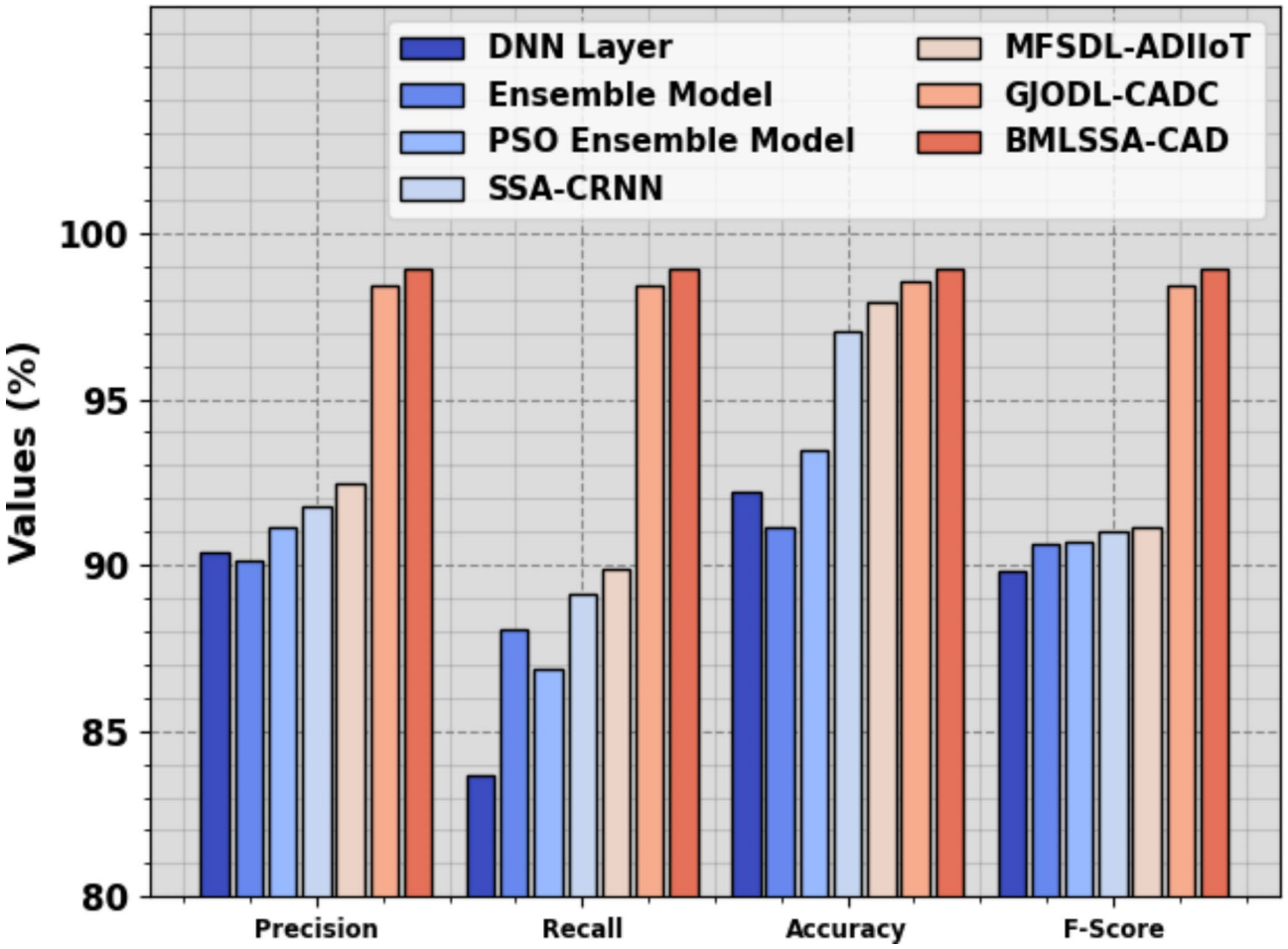

Fig. 18. Comparative analysis of BMLSSA-CAD technique on the UCI SECOM dataset.

Data availability

The datasets used and analyzed during the current study available from the corresponding author on reasonable request.

Received: 25 August 2024; Accepted: 11 November 2024
Published online: 26 November 2024

References

1. Alkahtani, H. & Aldhyani, T. H. Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms. *Complexity* **2021**, 5579851 (2021).
2. Alatawi, T. & Aljuhani, A. Anomaly detection framework in fog-to-things communication for industrial Internet of things. *Comput. Mater. Contin.* **73**, 1067–1086 (2022).

3. Kumar, R. & Tripathi, R. DBTP2SF: A deep blockchain-based trustworthy privacy-preserving secured framework in industrial internet of things systems. *Trans. Emerg. Telecommun Technol.* **32**, e4222 (2021).
4. Tomar, K., Bisht, K., Joshi, K. & Katarya, R. Cyber attack detection in IoT using deep learning techniques. In *Proceedings of the 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 3–4 March 2023* 1–6 (IEEE: Piscataway, NJ, USA, 2023).
5. Vaiyapuri, T., Sbai, Z., Alaskar, H. & Alaseem, N. A. Deep learning approaches for intrusion detection in IIoT networks–opportunities and future directions. *Int. J. Adv. Comput. Sci. Appl.* **12**, 86–92 (2021).
6. Hasan, T. et al. Securing industrial internet of things against botnet attacks using hybrid deep learning approach. *IEEE Trans. Netw. Sci. Eng.* **10**, 2952–2963 (2022).
7. AL-Nuaimi, B. T., Suhail, R. A. & El-kenawy, E. S. M. Adaptive feature selection based on machine learning algorithms for lung tumors diagnosis and the COVID-19 index. *J. Intell. Syst. Internet Things* **11**(2) (2024).
8. Li, F., Lin, J. & Han, H. F. S. L. Federated sequential learning-based cyberattack detection for Industrial Internet of things. *Ind. Artif. Intell.* **1**, 4 (2023).
9. Khan, F., Jan, M. A., Alturki, R., Alshehri, M. D., Shah, S. T. & Ur Rehman, A. A secure ensemble learning-based fog-cloud approach for cyberattack detection in IoMT. *IEEE Trans. Ind. Inf.* **19**, 10125–10132 (2023).
10. Alalayah, K. M. et al. Optimal deep learning based intruder identification in industrial internet of things environment. *Comput. Syst. Sci. Eng.* **46**, 3121–3139 (2023).
11. Saheed, Y. K., Abdulganiyu, O. H. & Tchakoucht, T. A. Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with Edge capabilities. *Appl. Soft Comput.* 111434 (2024).
12. Alani, M. M. & Awad, A. I. An Intelligent two-layer intrusion detection system for the internet of things. *IEEE Trans. Industr. Inf.* **19**(1), 683–692 (2022).
13. Golchha, R., Joshi, A. & Gupta, G. P. Voting-based ensemble learning approach for cyber attacks detection in industrial internet of things. *Procedia Comput. Sci.* **218**, 1752–1759 (2023).
14. Feng, X., Han, J., Zhang, R., Xu, S. & Xia, H. Security defense strategy algorithm for Internet of Things based on deep reinforcement learning. *High-Confidence Computing* **4**(1), 100167 (2024).
15. Awotunde, J. B. et al. An ensemble tree-based model for intrusion detection in industrial internet of things networks. *Applied Sciences*, **13**(4), 2479 (2023).
16. Ren, B. et al. A multiagents deep reinforcement learning autonomous security management approach for internet of things. *IEEE Internet Things J.* (2024).
17. Alattas, K. A. & Mardani, A. A novel extended internet of things (IoT) cybersecurity protection based on adaptive deep learning prediction for industrial manufacturing applications. *Environ. Dev. Sustain.* **24**(7), 9464–9480 (2022).
18. Xu, H., Sun, Z., Cao, Y. & Bilal, H. A data-driven approach for intrusion and anomaly detection using automated machine learning for the internet of things. *Soft. Comput.* **27**(19), 14469–14481 (2023).
19. Gondkar, S. R., Rv, S. B., Kavitha, S. & Gondkar, R. R. Sliced bidirectional gated recurrent unit with sparrow search optimizer for detecting the attacks in IoT environment. In *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* 1–7. (IEEE, 2024).
20. Harahsheh, K., Al-Naimat, R. & Chen, C. H. Using feature selection enhancement to evaluate attack detection in the internet of things environment. *Electronics* **13**(9), 1678 (2024).
21. Alshmrany, S. Innovative IoT threat detection: Weighted variational autoencoder-based hunter prey search algorithm for strengthening cybersecurity. *IETE J. Res.* 1–14 (2024).
22. Mohammed, I. H., Kumar, B. V., Babu, B. M., Goud, B. P. & Al-Attabi, K. Chaotic sparrow search algorithm with deep learning for anomaly detection in internet of things. In *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)* 1–6 (IEEE, 2023).
23. Arulkumar, V. et al. A novel cloud-assisted framework for consumer internet of things based on lanner swarm optimization algorithm in smart healthcare systems. *Multimedia Tools Appl.* 1–25 (2024).
24. Saheed, Y. K., Omole, A. I. & Sabit, M. O. GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data. *Sensors International* **6**, 100297 (2025).
25. Gaber, T., Awotunde, J. B., Folorunso, S. O., Ajagbe, S. A. & Eldesouky, E. Industrial internet of things intrusion detection method using machine learning and optimization techniques. *Wireless Communications and Mobile Computing*, **2023**(1), 3939895 (2023).
26. Altunay, H. C. & Albayrak, Z. A hybrid CNN + LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, An International Journal* **38**, 101322 (2023).
27. Wankhade, K. K., Dongre, S., Chandra, R., Krishnan, K. V. & Arasavilli, S. Machine learning-based detection of attacks and anomalies in industrial internet of things (IIoT) networks. In *International Conference on Applied Soft Computing and Communication Networks* 91–109 (Springer Nature Singapore, Singapore, 2023).
28. Altunay, H. C., Albayrak, Z., Özalp, A. N. & Cakmak, M. June. Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* 1–6 (IEEE, 2021).
29. Qaddoori, S. L. & Ali, Q. I. An efficient security model for industrial internet of things (IIoT) system based on machine learning principles. *Al-Rafidain Eng. J. (AREJ).* **28** (1), 329–340 (2023).
30. Altunay, H. C., Albayrak, Z. & Çakmak, M. Autoencoder-based intrusion detection in critical infrastructures. *Current Trends in Computing* **2**(1), 1–12 (2024).
31. Ellappan, V. et al. Sliding principal component and dynamic reward reinforcement learning based IIoT attack detection. *Scientific Reports* **13**(1), 20843 (2023).
32. Alani, M. M., Mauri, L. & Damiani, E. A two-stage cyber attack detection and classification system for smart grids. *Internet of Things* **24**, 100926 (2023).
33. Khadidos, A. O. et al. Binary Hunter–Prey Optimization with Machine Learning—Based Cybersecurity Solution on Internet of Things Environment. *Sensors*, **23**(16), 7207 (2023).
34. Khan, M. A., Naveed, Q. N., Lasisi, A., Kaushik, S. & Kumar, S. A multi-layered assessment system for trustworthiness enhancement and reliability for industrial wireless sensor networks. *Wireless Pers. Commun.* **137** (4), 1997–2036 (2024).
35. Alwasel, B., Aldribi, A., Alreshoodi, M., Alsukayti, I. S. & Alsuhaibani, M. Leveraging graph-based representations to enhance machine learning performance in IIoT network security and attack detection. *Applied Sciences* **13**(13), 7774 (2023).
36. Zhang, Y. et al. July. An efficient CNN+ sparse transformer-based intrusion detection method for IoT. In *International Conference on Intelligent Computing* 482–493 (Springer Nature Singapore, Singapore 2024).
37. Pundir, S. et al. MADP-IIME: Malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach. *Multimedia Syst.* **29** (3), 1785–1797 (2023).
38. Ghasemkhani, B. et al. Federated multi-label learning (FMLL): Innovative method for classification tasks in animal science. *Animals* **14**(14), 2021 (2024).
39. Alrowais, F. et al. Automated machine learning enabled cybersecurity threat detection in internet of things environment. *Comput. Syst. Sci. Eng.*, **45**(1). (2023).
40. Tiwari, R. S., Lakshmi, D., Das, T. K., Tripathy, A. K. & Li, K. C. A lightweight optimized intrusion detection system using machine learning for edge-based IIoT security. *Telecommunication Syst.* 1–20 (2024).

41. Deepa, B. & Ramesh, K. Epileptic seizure detection using deep learning through min max scaler normalization. *Int. J. Health Sci.* **6**, 10981–10996 (2022).
42. Zhang, Q. Optimization of nonlinear convolutional neural networks based on improved chameleon group algorithm. *Scalable Computing: Pract. Experience*. **25**(2), 840–847 (2024).
43. Ahmad, M., Tang, X. W., Qiu, J. N. & Ahmad, F. Evaluating seismic soil liquefaction potential using bayesian belief network and C4. 5 decision tree approaches. *Applied Sciences* **9**(20), 4226 (2019).
44. Zhang, J., Zhu, X. & Li, J. Intelligent path planning with an improved sparrow search algorithm for workshop UAV inspection. *Sensors* **24**(4), 1104 (2024).
45. Maghrabi, L. A. et al. Golden jackal optimization with a deep learning-based cybersecurity solution in industrial internet of things systems. *Electronics*, *12*(19), 4091 (2023).

Acknowledgements
The authors thank the CICHE Research Center and SISAu Research Group for supporting this work. The results of this work are part of the project "Tecnologías de la Industria 4.0 en Educación, Salud, Empresa e Industria" developed by Universidad Tecnológica Indoamérica.

Author contributions
All authors have same contribution.

Funding
Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to J.V.-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.