scientific reports

Check for updates

OPEN

An effectiveness of deep learning with fox optimizer-based feature selection model for securing cyberattack detection in IoT environments

Mimouna Abdullah Alkhonaini

The fast development of Internet of Things (IoT) tools in smart cities has presented many advantages, improving sustainability, automation, and urban efficiency. Still, these interlinked systems further pose critical cybersecurity difficulties, including cyberattacks, data breaches, and unauthorized access that may compromise essential frameworks. Usually, cybersecurity is considered a group of processes and technologies intended to safeguard networks, computers, data, and programs against malicious attacks, harm, activities, or unauthorized access. IoT cybersecurity targets to minimize cybersecurity threats for users and organizations regarding the safety of IoT assets and confidentiality. Novel cybersecurity technologies are continually developing and give opportunities and challenges to IoT cybersecurity organizations. Deep learning (DL) is one of the main technologies of today's smart cybersecurity policies or systems for functioning intelligently. This paper presents a Fox Optimizer-Based Feature Selection with Deep Learning for Securing Cyberattack Detection (FOFSDL-SCD) model. This paper aims to analyze cybersecurity-driven approaches for enhancing IoT networks' resilience and threat detection capabilities using advanced techniques. Initially, the data pre-processing stage utilizes the min-max normalization method to transform the input data into a beneficial system. Furthermore, the FOFSDL-SCD model utilizes the Fox optimizer algorithm (FOA) method for the feature selection process to select the most significant features from the dataset. Moreover, the temporal convolutional network (TCN) model is employed for classification. Finally, the dung beetle optimization (DBO)-based hyperparameter selection method is performed to improve the classification outcomes of the TCN model. The performance validation of the FOFSDL-SCD approach is examined under the Edge-IIoT dataset. The comparison study of the FOFSDL-SCD approach demonstrated a superior accuracy, precision, recall, and F1-Score of 99.38%, 96.27%, 96.26%, and 96.27% over existing models.

Keywords Cyberattack detection, Cybersecurity, IoT, Feature selection, Deep learning, Dung beetle optimization, Data Pre-processing

The IoT has become a vital technology, carrying significant consequences for cybersecurity. IoT systems are widespread, connected, and frequently lack basic security mechanisms, exposing them to cyber risks[1]. Hackers can use these flaws to access private information, carry out distributed denial-of-service (DDoS) attacks, and capture vital systems controllers. A massive cyberattack on IoT systems could lead to serious effects, such as interrupting critical operations and causing significant financial loss[2]. These IoT devices share sensor information through an edge device or gateway. This data is uploaded to the cloud for inspection or handled on-site. From time to time, these devices share information and respond accordingly[3]. Generally, IoT devices operate independently without direct human input. The IoT is a fast-growing area that brings specific difficulties related to compatibility, data protection, and security. Information from IoT devices is at risk of cyberattacks due to their limited power usage, low computing ability, and restricted memory[4]. These issues arise from the distributed setup of IoT systems. Hence, a strong security plan is necessary. Present security tools like intrusion detection systems (IDSs) and firewalls may not be effective in safeguarding the IoT environment, as

Department of Computer Science, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia. email: mkhonaini@psu.edu.sa

IoT demands protection with high performance, adaptability, and speed provided by the Internet[5]. ID plays a vital role in protecting organizational IoT setups, serving as a preventive step to identify malicious entities or unauthorized access in an IoT infrastructure[6]. Reliable, ideal, and expandable ID methods are needed to guard IoT systems against recognized and new attack forms[7]. This task becomes challenging, particularly in the case of IoT environments, due to the distinctiveness and changing nature of devices, transmission methods, and emerging risks in the network. Figure 1 illustrates the overview of cybersecurity challenges in IoT, highlighting data flow between devices, edge gateways, and the cloud.

Conventional IDSs are unsuitable for handling IoT systems' wide and increasing scale. Technology is becoming increasingly vital daily, which means cybersecurity and cybercrime tools advance across the entire production sector, demanding spending on cybersecurity defences. Meanwhile, advanced methods are established for IoT security control[8]. Moreover, cyberattacks on smart grids, as key framework components, are particularly vulnerable and involve higher expenses, severely impacting the protection of authorities and the public. Due to their strong results in various prediction-related fields, currently, researchers have concentrated on DL and machine learning (ML) models[9]. Applying AI models such as ML and DL techniques may present useful methods for handling data to detect and forecast potential cyber risks. DL methods identify cyber risks that are gaining popularity more rapidly than older methods, making counteraction more successful[10]. DL methods have become a steadily standard tool in cybersecurity, quickly becoming key to strong defence tactics against malicious intrusions. The proliferation of IoT devices has expanded attack surfaces, making them prime targets for cyber threats. Conventional security measures mostly fall short due to the limited computational resources of these devices. Consequently, there is a pressing requirement for advanced IDSs capable of operating efficiently within these constraints. DL models present promising solutions by enabling the detection of complex attack patterns. However, the high dimensionality of IoT data can affect performance. Feature selection methods significantly detect the most relevant attributes, improve detection accuracy, and reduce computational load. Integrating robust feature selection with DL techniques can substantially improve the efficiency of IDS in IoT environments.

This paper presents a Fox Optimizer-Based Feature Selection with Deep Learning for Securing Cyberattack Detection (FOFSDL-SCD) model. This paper aims to analyze cybersecurity-driven approaches for enhancing IoT networks' resilience and threat detection capabilities using advanced techniques. Initially, the data pre-processing stage utilizes the min-max normalization method to transform the input data into a beneficial system. Furthermore, the FOFSDL-SCD model utilizes the Fox optimizer algorithm (FOA) method for the feature selection process to select the most significant features from the dataset. Moreover, the temporal convolutional

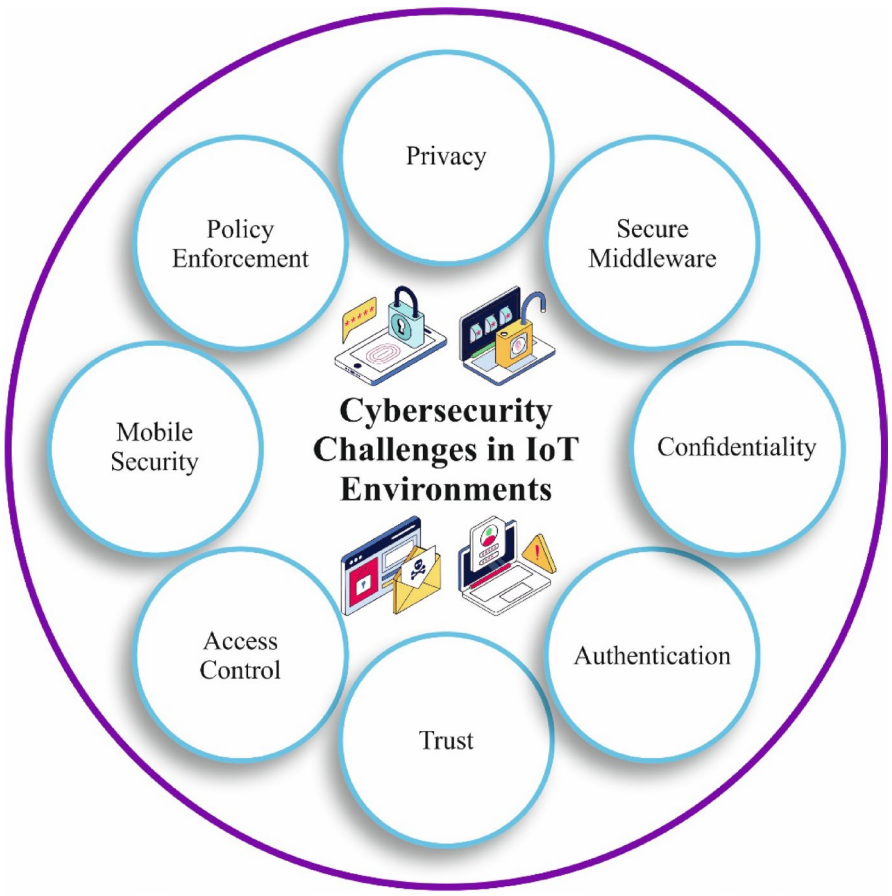

Fig. 1. Overview of cybersecurity challenges in IoT environments.

network (TCN) model is employed for classification. Finally, the dung beetle optimization (DBO)-based hyperparameter selection method is performed to improve the classification outcomes of the TCN model. The performance validation of the FOFSDL-SCD approach is examined under the Edge-IIoT dataset. The key contribution of the FOFSDL-SCD approach is listed below.

- The FOFSDL-SCD model utilizes min-max normalization to scale input features uniformly, enhancing data consistency and helping the model learn more effectively. This step improves overall training stability and contributes to better detection performance. Standardizing the data range assists in efficient processing within the model.
- The FOFSDL-SCD method employs the FOA technique to select the most relevant features, minimizing redundant data and enhancing model efficiency. This optimization results in improved detection accuracy by concentrating on critical inputs. It assists faster processing and better overall performance in cyberattack detection.
- The FOFSDL-SCD approach implements the TCN technique to classify cyberattacks by capturing temporal dependencies in IoT data, enhancing pattern recognition over time. This approach improves the model's capability of accurately detecting sequential attack behaviours. It ensures robust performance in dynamic and time-sensitive environments.
- The FOFSDL-SCD methodology applies the DBO method for hyperparameter tuning to improve the model's convergence and learning efficiency. This optimization refines parameters for better detection accuracy and stability, strengthening the model's overall performance in complex IoT environments.
- The FOFSDL-SCD technique integrates FOA-based feature selection, TCN classification, and DBO hyperparameter tuning to form a novel and adaptive framework that effectually balances precision, efficiency, and optimization. This unique integration improves cyberattack detection by dynamically adapting to IoT data characteristics. It presents a robust solution tailored specifically for the growing challenges of IoT security environments.

The article is structured as follows: Section "Literature review" presents the literature review, Section "The proposed methodology" outlines the proposed method, Section "Performance Validation" details the results evaluation, and Section "Conclusion" concludes the study.

Literature review

Ragab et al.[11] designed a next-generation cybersecurity attack detection through an ensemble DL (NGCAD-EDL) paradigm within the IIoT setting. The motive is to recognize cyberattacks automatically. A data normalization process using min-max normalization is conducted. Then, the honey-badger algorithm (HBA) methodology chooses the feature subsets. Ghosh[12] introduced a new framework named Automatic Separate Guided Attention Federated Graph Neural Network (ASGAFGNN) to predict and detect cyber threats. Primarily, cyberattack information is collected and pre-processed to improve the features. The pre-processed information afterwards underwent feature extraction to obtain temporal, local, and global features utilizing a hybrid vision transformer and bidirectional LSTM. Knieps[13] focused on the network structure of cybersecurity control, considering the interaction among the current EU cybersecurity control design. Inuwa and Das[14] explored the rising difficulties of cybersecurity, which became progressively vulnerable to cyberattacks. The extensive usage of IoT devices strengthens the intricate communications among devices and intensifies the data traffic, producing numerous chances for cyber enemies. Therefore, identifying and countering cyberattacks targeting IoT systems has become a complex necessity in cybersecurity. Yang[15] presented a fresh method for cybersecurity study, which uses the blockchain (BC) technique for fuzzy ML and cloud computing (CC). The smart grid combined CC technique observes and communicates information through electric vehicles. However, the fuzzy adversarial Q-stochastic (FAQS) method assesses possibly risky activities. Xu et al.[16] suggested a data-driven intrusion and anomaly detection paradigm. An automated ML model is also utilized to detect the process by auto-tuned hyperparameters, which is ideal for data classification. Manickam et al.[17] recommended a novel billiard-based optimizer with DL-assisted AD and classification (BBODL-ADC) model in IoT infrastructure. The model aims to detect and classify anomalies in the IoT environment appropriately. This model employs a binary pigeonhole optimization (BPO) framework for successful FS. Also, this method deploys an Elman RNN (ERNN) strategy for anomaly detection and classification. Furthermore, the BBO method could select parameters through the ERNN framework.

Aboalela et al.[18] developed an efficient and accurate method for detecting Distributed Denial of Service (DDoS) cyberattacks in IoT environments using advanced feature selection, DL models, and optimized hyperparameter tuning. Ghadi et al.[19] explored federated learning (FL) integration with IoT across domains like smart cities, healthcare, and transportation, highlighting its decentralized AI benefits and addressing critical security challenges in FL-IoT systems. Alrayes et al.[20] proposed an automated and accurate cybersecurity detection method for Industrial IoT environments using advanced normalization, feature selection, DL, and hyperparameter optimization techniques. Ghadi et al.[21] explored the utilization of artificial intelligence (AI) and big data (BD) models for detecting and mitigating cybersecurity threats in smart grids by analyzing diverse attack types and addressing AI-related challenges. Dhanvijay, Dhanvijay, and Kamble[22] improved intrusion detection in IoT networks by integrating advanced data pre-processing, optimized feature selection, DL ensembles, and prediction scoring to improve accuracy and reduce false positives. Khan et al.[23] investigated advanced antenna designs and materials for improving communication efficiency, adaptability, and performance in IoT applications across various industries. Alkhalifa et al.[24] improved intrusion detection in IoT networks by incorporating dimensionality reduction, DL classification, and hyperparameter tuning for enhanced cybersecurity performance. Mazhar et al.[25] explored the integration of smart grids with Industry 5.0 to improve

industrial efficiency, resource optimization, and environmental sustainability. Alkahtani et al.[26] proposed a model to improve IoT cybersecurity by integrating advanced ensemble learning models with metaheuristic optimization for efficient attack detection and classification performance. Mazhar et al.[27] explored how ML and DL techniques can enhance security in IoT systems by detecting cyberattack patterns and protecting devices from growing threats. Adeke et al.[28] examined how feature selection affects adversarial attack transferability across ML and DL-based intrusion detection systems to improve their robustness against black-box attacks.

Despite notable advancements, various limitations and a clear research gap still exist. Most models emphasize FS and DL but overlook real-time adaptability in dynamic IIoT contexts. Many techniques lack scalability in high-traffic IoT environments. FL, BC, and CC integration remain underexplored, specifically in handling adversarial attacks. Few works address the interpretability of DL decisions, which restricts trust in critical infrastructures. A research gap exists in developing lightweight, energy-efficient IDS tailored for constrained IoT devices. Cross-domain generalization and model robustness under unseen cyberattack vectors also require focused exploration.

The proposed methodology

This article proposes the FOFSDL-SCD method. This paper analyses cybersecurity-driven techniques for improving IoT networks' resilience and threat detection abilities utilizing advanced methods. Data pre-processing, dimensionality reduction using FOX, classification, and parameter tuning are required. Figure 2 indicates the entire workflow of the FOFSDL-SCD model.

Pre-processing through normalization

At the primary step, the data pre-processing stage utilizes the min-max normalization method to transform the input data into a beneficial system. Min-max normalization is a data scaling method employed to convert features to a secure range, usually [0,1], conserving the relationships between original data values[29]. Normalization safeguards uniformity across features in cybersecurity-driven IoT networks, where network traffic and sensor data frequently differ extensively in scale. This aids ML methods in noticing anomalies more efficiently by decreasing bias toward features with greater arithmetical ranges. Applying this normalization improves the precision and consistency of intrusion detection methods in IoT systems. The mathematical formulation is given below in Eq. (1).

$$Y = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

Here, X signifies the original data, Y epitomizes the normalized data, and $X_{\max}$ and $X_{\min}$ denote the maximum and minimum values, respectively.

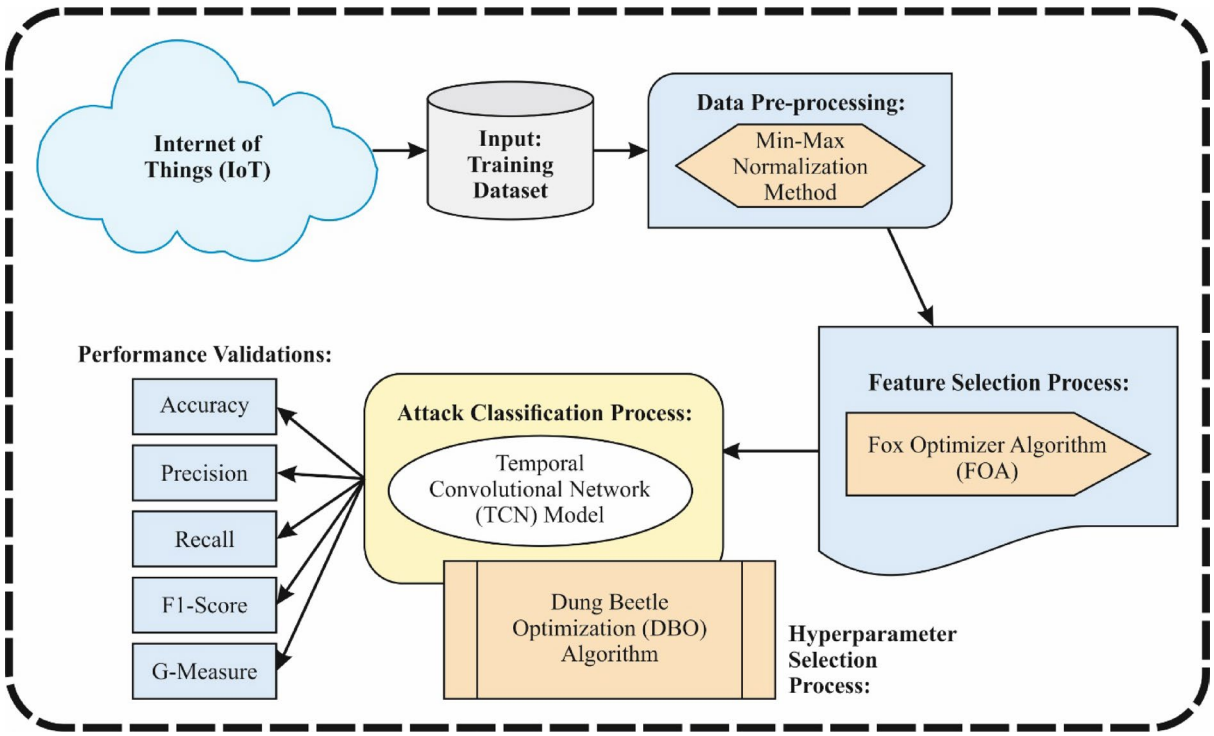

Fig. 2. Entire workflow of the FOFSDL-SCD method.

FOA-based feature selection method

Besides, the proposed FOFSDL-SCD designs FOA for the FS procedure to select the most significant features from the dataset. FOX is a new optimizer model stimulated by the red fox's predatory behaviour[30]. The FOX is selected according to its progressive mechanisms, which deal with the restrictions of present swarm-based methods and achieve an improved balance between the exploitation and exploration stages. This mathematically verified performance highlights FOX's adaptability, applicability, and robustness to composite optimizer tasks containing reinforcement learning, whereas practical exploitation and exploration are essential. Still, FOX intends to recognize the best solution by assessing the optimal fitness values through the searching agent's population. The FOX works over numerous iterations with many agents, all searching for the best value of fitness (improved solution). It includes dual main steps: exploration and exploitation. During the exploration step, agents use the random walking approach to find possible solutions for how red foxes look for their victims. They utilize their capability to identify ultrasound signals to help with this hunt. During this stage, the agents approximate the distance to their target depending on the time the ultrasound signal takes to reach them. Therefore, FOX uses a different model for measurements, whereas agents jump after their approximation; they might take the prey according to the lapse time of the sound signal. Therefore, the agent's success in taking prey is carefully related to its capability to understand the sound's travelling time while jumping precisely.

$$s = BestPosition_i \times T_i \tag{2}$$

The distance of FA from its prey (DFP) is described as demonstrated:

$$DFP_i = DSF_i \times 0.5 \tag{3}$$

After the FA fixes the distance to its target, it should instantly jump to capture it. The requisite jumping height was calculated utilizing the succeeding Eq. (4):

$$jump_i = 0.5 \times 9.81 \times t^2 \tag{4}$$

The FA travels to a novel location in the exploitation and exploration phases. The novel location is verified during the exploitation phase using the following Eqs. (5) and (6):

$$X_{i+1} = DFP_i \times jump_i \times c_1 \tag{5}$$

$$X_{i+1} = DFP_i \times jump_i \times c_2 \tag{6}$$

The c_1 and c_2 values are 0.180 0.820, individually estimated according to the jumping dynamics of the FA. The jump agents must move both toward the opposite or the northeast direction. For the exploration phase, the novel location is computed utilizing the succeeding Eq. (7):

$$X_{i+1} = BestX_i \times rand\,(1,\ dim) \times \min\,(tt) \times a \tag{7}$$

Now, tt refers to the average time, equivalent to the vector amount T divided by the problem size. The FOX uses a stationary trade-off approach, balancing the exploitation and exploration phases at each 0.5. During the exploration stage, the optimizer imitates the detection abilities of the fox's target over random walks. This model mimics how the fox can seek a target in its surroundings, allowing the optimizer to discover possible solutions. This stage intends to converge on the best solution using the most positive regions recognized in the exploration stage.

The fitness function (FF) determines the classification precision and the chosen feature amounts. It maximizes the classification precision and lowers the set size of the designated attributes. Then, the succeeding FF is applied to evaluate individual solutions, as presented in Eq. (8).

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{8}$$

Here, $ErrorRate$ denotes the classification $ErrorRate$ using the chosen features. $ErrorRate$ is measured as the incorrect percentage classified to the number of classifications completed, specified as the value among (0,1). $\#SF$ represents selected feature counts, and $\#All\_F$ is the comprehensive quantity of features in the new dataset. α is utilized to control the significance of subset length and classification quality.

Classification using the TCN model

The TCN method is deployed for the classification process. TCN addresses the task of acquiring either local or long-term dependency in the network data by employing dilated and causal convolution[31]. Various recurring techniques depend upon sequential processing, while the TCN utilizes causal convolution to maintain the temporal data order, guaranteeing that upcoming data is not employed to forecast preceding events. Furthermore, dilated convolution increases the receptive area without rising parameter counts, allowing the technique to acquire longer-range dependencies more effectively. This makes TCN efficient in handling either long-term patterns or short-term variations. This integration of dilated and causal convolutions aids in enhancing the performance of the model. The convolutional module successfully takes the local time dependency in the input data over a convolutional operation. Conventional RNN contains superior computational efficacy and enhanced parallelization proficiencies while processing long time-series data. This model comprises dilated and causal

convolutions and residual links. Dilated convolution increases the receptive area without dropping resolution; causal convolution safeguards the technique about the temporal sequence of the data, and residual connection aids in reducing the issue of gradient vanishing. Assume that the sequence of input $Z = [z_1, z_2, \ldots, z_n]$; here, z_i refers to the output of the self-attention module.

$$Y_t = \sum_{k=0}^{K-1} W_k \cdot Z_{t-k} \tag{9}$$

W_k signifies convolution kernel weight, K represents the size of the convolution kernel, and Y_t is an output of tth time-step. Over causal convolution, only input data of t is guaranteed to be employed at every time-step t.

This method enables the technique to deal with long-time dependency by improving the parameter counts:

$$Y_t = \sum_{k=0}^{K-1} W_k \cdot Z_{t-d \cdot k} \tag{10}$$

d refers to the dilation coefficient regulating the convolution kernel's hole size. This model employs residual links to mitigate the issues of gradient explosion and disappearance. This method permits the technique to bypass definite layers, assisting the data flow and gradient.

$$Y_t = F(Z_t) + Z_t \tag{11}$$

Z_t refers to the input, and $F(Z_t)$ indicates the output after dilated and causal convolution.

The convolutional time-series module is formed by many dilated and causal convolutional layers with every residual connection. This loaded framework acquires layer-wise dependencies of diverse time scales, thus enhancing the modelling proficiency of longer time-series data. Letting an output of the lth layer is Y^l. The mathematical model is:

$$Y^l = ReLU\left(LN\left(F^l\left(Z^l\right)\right)\right) + Z^l \tag{12}$$

Where LN represents layer normalization, $ReLU$ refers to the activation function, Z^l signifies the input of this layer, and F^l indicates the convolution operation of the lth layer. Within the previous layer of convolutional modelling, a fully connected (FC) layer can be employed to modify an output of the convolution process into a last value.

$$\widehat{Y} = OW + b \tag{13}$$

Now, b indicates a biased term, and W represents a weight matrix. The FC layer could map an output of a higher-dimensional convolution to the needed dimension for attaining the outcome. Over these operations, this approach can effectively take a long time and local dependencies in input data. Concurrently, the generalizability of the training method is further enhanced by over-optimization approaches like early stopping, dropout, and batch normalization. The output is integrated to present an effective and comprehensive representation. Figure 3 illustrates the framework of the TCN method.

DBO-based hyperparameter selection approach

Finally, the DBO-based hyperparameter selection method is implemented to improve the classification outcomes of the TCN model. The elementary DBO model is naturally stimulated by the foraging, dancing, stealing, breeding, and rolling behaviours of DBs[32]. Based on these behaviours, four population-updated tactics are designed.

Rollerball dung beetles

Naturally, DBs utilize solar navigation to manage a straight route while rolling their dung balls. Equation (14) is applied to change the location of the rolling DB:

$$\begin{gathered} x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \\ \Delta x = |x_i(t) - X^{\omega}| \end{gathered} \tag{14}$$

Whereas t characterizes the present iteration counts, and $x_i(t)$ exemplifies the place of the ith DB at the tth iteration. α specifies whether the DB deviates from its early route, with its value defined randomly as 1 and -1. $k \in (0,0.2]$ means constant that denotes the coefficient of deflection, and b refers to constant using the value range of $(0,1)$. X^{ω} signifies the global poor position, and Δx is applied to mimic solar light. If the DB encounters a problem and can no longer continue rolling, it should dance to decide its new rolling path. This behaviour of dancing is outlined as demonstrated:

$$x_i(t+1) = x_i(t) + tan(\theta)\,|x_i(t) - x_i(t-1)| \tag{15}$$

Here, $\theta \in [0, \pi]$, and the location is not upgraded if θ captures values like 0, $\pi/2$, and π.

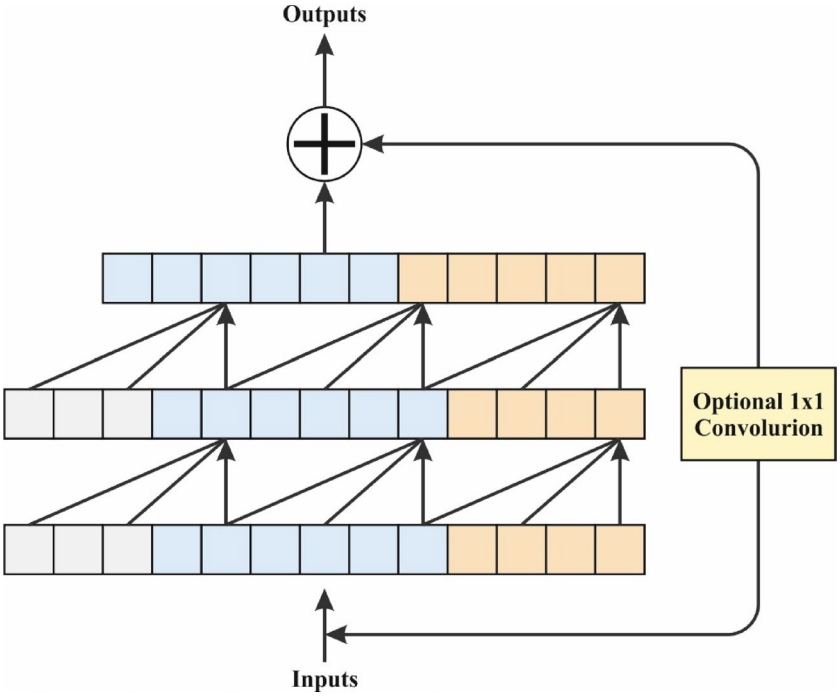

Fig. 3. Architecture of the TCN method.

Breeding dung beetles
To guarantee a secure background for their offspring, DBs move the dung balls to safe places and hide them before laying their eggs. Equation (16) presents a boundary selection approach to mimic the egg-laying zone of female DBs.

$$\begin{aligned} Lb^* &= \max\left(X^* \times \left(1-R\right),\ Lb\right) \\ Ub^* &= \min\left(X^* \times \left(1+R\right),\ Ub\right) \end{aligned} \tag{16}$$

On the other hand, X^* embodies the local best value, and Ub^* and Lb^* characterize the upper and lower limits of the spawning space. $R = 1 - T/T_{\max}$ and $T_{\max}$ symbolize the maximal iteration counts, and Ub and Lb imply the upper and lower limits of the optimizer issue, correspondingly. According to the equation mentioned above, the borders of the egg-laying zone are dynamically established by the R variation. Therefore, the DB's breeding location is upgraded constantly, as stated by the succeeding mathematical representation:

$$B_i\left(t+1\right) = X^* + b_1 \times \left(B_i\left(t\right) - Lb^*\right) + b_2 \times \left(B_i\left(t\right) - Ub^*\right) \tag{17}$$

Here, $B_i\left(t\right)$ signifies the locality of the ith area at the tth iteration. b_1 and b_2 denote two self-governing randomly formed vectors, all using the dimension of 1x D, whereas D embodies the dimensionality.

Foraging dung beetles
Naturally, while foraging, DBs prefer a safer place in a way equivalent to after they lay their eggs. The unique description of this safer place is re-examined and is characterized by the succeeding Eq. (18):

$$\begin{aligned} Lb^b &= \max\left(X^b \times \left(1-R\right),\ Lb\right) \\ Ub^b &= \min\left(X^b \times \left(1+R\right),\ Ub\right) \end{aligned} \tag{18}$$

Here, X^b exemplifies the global finest place, and $Ub^b$and Lb^b denotes the upper and lower boundaries of the optimum foraging zone, correspondingly. Then, the place of the more minor DB is upgraded as shown:

$$x_i\left(t+1\right) = x_i\left(t\right) + C_1 \times \left(x_i\left(t\right) - Lb^b\right) + C_2 \times \left(x_i\left(t\right) - Ub^b\right) \tag{19}$$

C_1 refers to a random variable that follows a standard distribution, and C_2 signifies an arbitrary variable in the interval of $(0,1)$.

Stealing dung beetles
This behaviour involves stealing dung balls from another beetle. In the iterative procedure, the location-updated mechanism for the stealing beetle is directed by Eq. (20).

$$x_i(t+1) = X^b + S \times g \times \left(|x_i(t) - X^*| + \left|x_i(t) - X^b\right|\right) \tag{20}$$

Meanwhile, S signifies a constant, and g denotes a randomly generated vector with dimensions following the standard distributions. Algorithm 1 represents the DBO model.

The DBO method obtains an FF to accomplish enhanced classification performance. It governs a progressive number to epitomize the higher performance of the candidate solutions. The minimization of the classification error rate is measured as the FF, as provided in Eq. (21).

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{no\ of\ misclassified\ samples}{Total\ no\ of\ samples} \times 100 \end{aligned} \tag{21}$$

Performance validation

The performance analysis of the FOFSDL-SCD model is examined under the Edge-IIoT dataset[33]. This dataset comprises 36,000 records under 12 classes, as depicted in Table 1. The complete number of features is 63, but only 27 are selected.

Figure 4 depicts the classifier results of the FOFSDL-SCD model. Figure 4a and b demonstrates the confusion matrices with precise detection and classification of all classes under 70% and 30%. Figure 4c shows the PR inspection, specifying maximum performance in all class labels. Ultimately, Fig. 4d explains the ROC investigation, determining capable outcomes with higher ROC values for individual classes.

Table 2; Fig. 5 institute the attack recognition of the FOFSDL-SCD model at 70%:30%. The result specified that the FOFSDL-SCD approach accurately identified all the different class labels. Under 70% TRPHE, the proposed FOFSDL-SCD approach achieves an average $accu_y$ of 99.38%, $prec_n$ of 96.27%, $reca_l$ of 96.26%, $F1_{score}$ of 96.27%, and $G_{Measure}$ of 96.27%. Moreover, under 30% TSPHE, the proposed FOFSDL-SCD approach achieves an average $accu_y$ of 99.36%, $prec_n$ of 96.16%, $reca_l$ of 96.18%, $F1_{score}$ of 96.17%, and $G_{Measure}$ of 96.17%.

Figure 6 shows the training (TRNG) $accu_y$ and validation (VALID) $accu_y$ outcomes of the FOFSDL-SCD approach. Both values are calculated for 0–25 epochs. The figure emphasized that both $accu_y$ Values demonstrate an increasing propensity, indicating the FOFSDL-SCD approach's efficiency with upgraded performance across numerous iterations. Moreover, both $accu_y$ remain close to the epochs, reflecting minimal overfitting and revealing the FOFSDL-SCD approach's improved performance.

Figure 7 presents the TRNG loss and VALID loss graphs of the FOFSDL-SCD approach. Both values are calculated for 0–25 epochs. It is symbolized that both values elucidate a declining propensity, reporting the FOFSDL-SCD model's effectiveness in equalizing a trade-off. The persistent decline further promises the FOFSDL-SCD approach's enhanced performance.

The comparison exploration of the FOFSDL-SCD model with present methods is exhibited in Table 3; Fig. 8[34,35]. Under $accu_y$, the FOFSDL-SCD model has a greater $accu_y$ of 99.38%, whereas LSTM-KPCA, Stacked Unsupervised FL, FL-MA, Generic CNN, Xception, VGG16, and InceptionResnetV2 methodologies get low $accu_y$ of 98.00%, 98.00%, 99.20%, 92.10%, 92.10%, 97.69%, and 88.37%, correspondingly. Likewise, under $prec_n$, the FOFSDL-SCD model got the highest $prec_n$ of 96.27%. In contrast, LSTM-KPCA, Stacked Unsupervised FL, FL-MA, Generic CNN, Xception, VGG16, and InceptionResnetV2 methodologies have gotten a lower $prec_n$ of 91.00%, 88.00%, 95.10%, 90.20%, 89.76%, 89.79%, and 84.35%, respectively. Finally, under $F1_{score}$. The FOFSDL-SCD model has a higher $F1_{score}$ of 96.27%, while LSTM-KPCA, Stacked Unsupervised FL, FL-MA, Generic CNN, Xception, VGG16, and InceptionResnetV2 methodologies get low $F1_{score}$ of 87.00%, 90.00%, 96.02%, 90.79%, 90.22%, 87.41%, and 85.22%, correspondingly. This demonstrates

Edge-IIoT Dataset	
Class Labels	**Data 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
Fingerprinting	3000
Total Record	36,000

Table 1. Details of Edge-IIoT dataset.

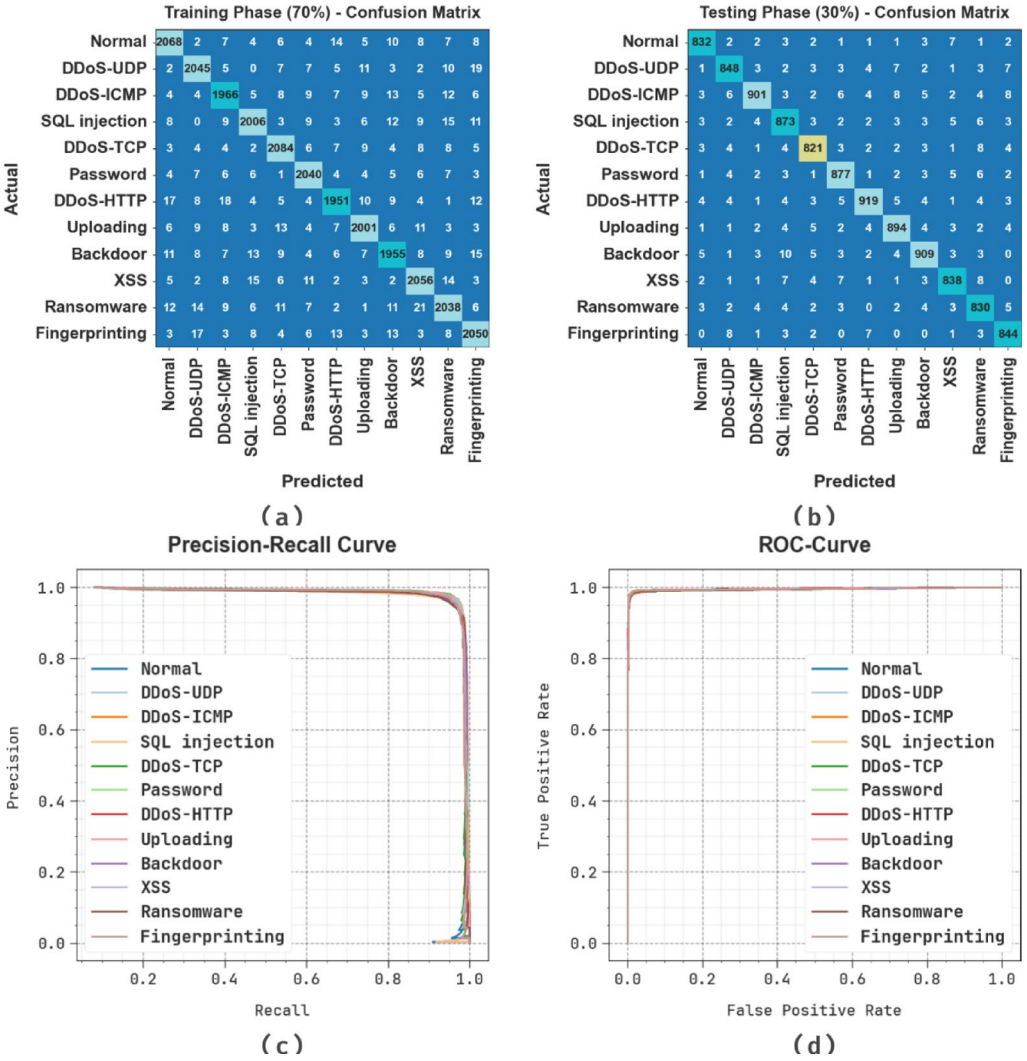

Fig. 4. Classifier outcome of (**a**-**b**) 70% and 30% confusion matrices, (**c**) curve of PR, and (**d**) curve of ROC.

Class Labels	$Accu_y$	$Prec_n$	$Recal_l$	$F1_{score}$	$G_{Measure}$
TRPHE (70%)					
Normal	99.40	96.50	96.50	96.50	96.50
DDoS-UDP	99.42	96.46	96.64	96.55	96.55
DDoS-ICMP	99.34	95.90	96.00	95.95	95.95
SQL injection	99.40	96.81	95.93	96.37	96.37
DDoS-TCP	99.47	96.62	97.20	96.91	96.91
Password	99.51	96.64	97.47	97.05	97.05
DDoS-HTTP	99.36	96.54	95.50	96.01	96.02
Uploading	99.44	96.71	96.48	96.60	96.60
Backdoor	99.27	95.69	95.27	95.48	95.48
XSS	99.38	96.03	96.66	96.34	96.35
Ransomware	99.23	95.59	95.32	95.46	95.46
Fingerprinting	99.32	95.75	96.20	95.97	95.97
Average	99.38	96.27	96.26	96.27	96.27
TSPHE (30%)					
Normal	99.53	96.97	97.08	97.03	97.03
DDoS-UDP	99.34	96.04	95.93	95.98	95.98
DDoS-ICMP	99.31	97.41	94.64	96.00	96.01
SQL injection	99.23	94.89	96.04	95.46	95.46
DDoS-TCP	99.38	96.25	95.91	96.08	96.08
Password	99.40	96.16	96.69	96.43	96.43
DDoS-HTTP	99.39	97.04	96.03	96.53	96.53
Uploading	99.38	96.23	96.54	96.39	96.39
Backdoor	99.32	96.39	95.89	96.14	96.14
XSS	99.38	96.32	95.99	96.16	96.16
Ransomware	99.26	94.53	96.29	95.40	95.41
Fingerprinting	99.42	95.69	97.12	96.40	96.40
Average	99.36	96.16	96.18	96.17	96.17

Table 2. Attack detection of FOFSDL-SCD model under 70%:30%.

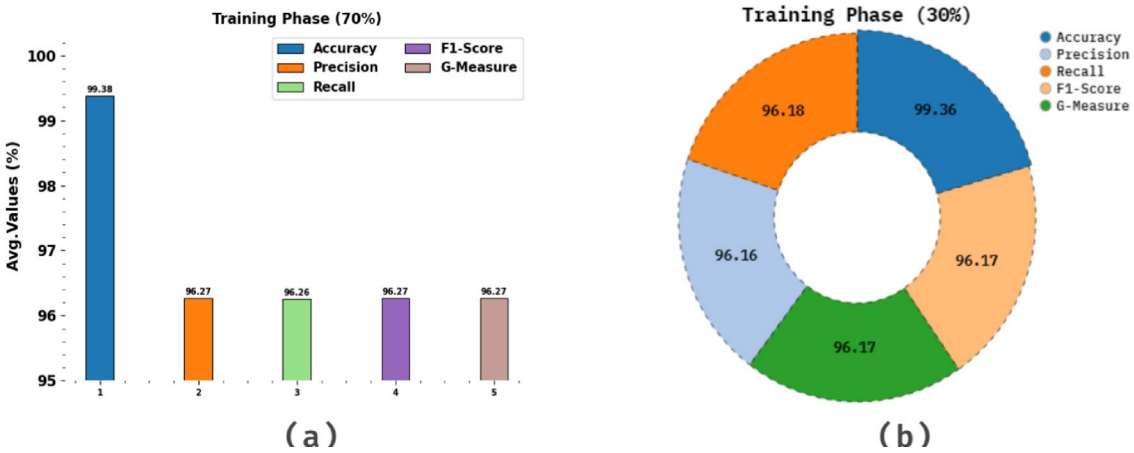

Fig. 5. Average values of FOFSDL-SCD model (a) 70% TSPHE and (b) 30%TRPHE.

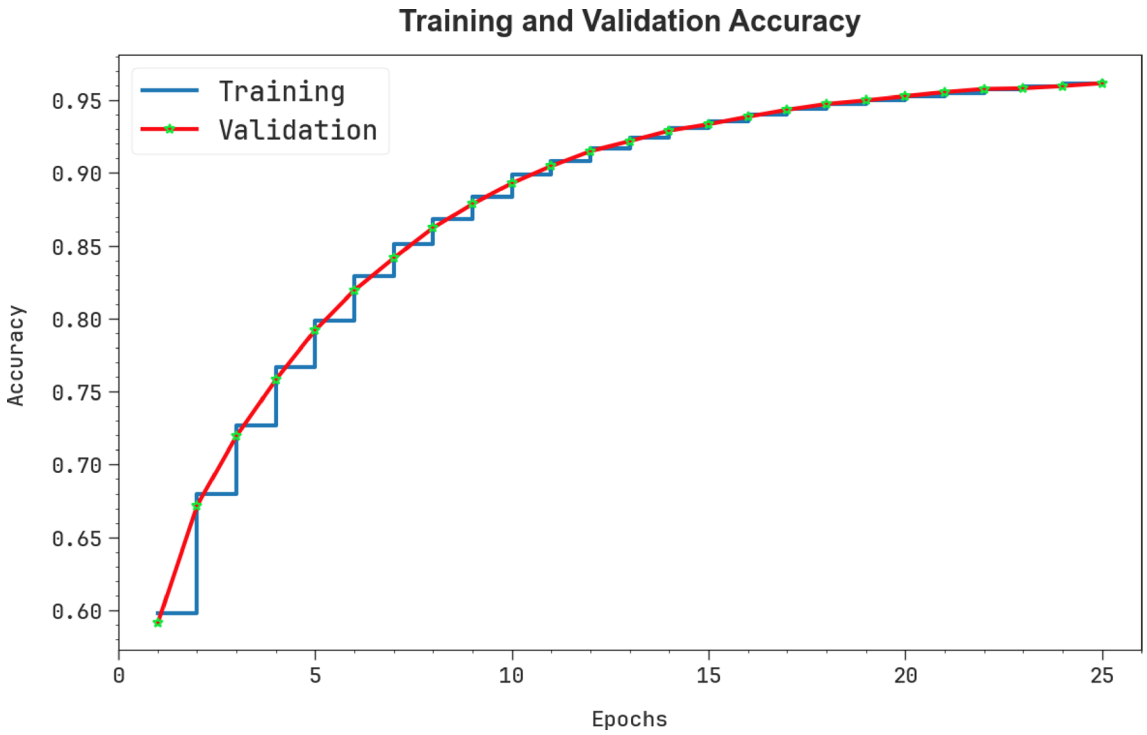

Fig. 6. $Accu_y$ curve of FOFSDL-SCD method.

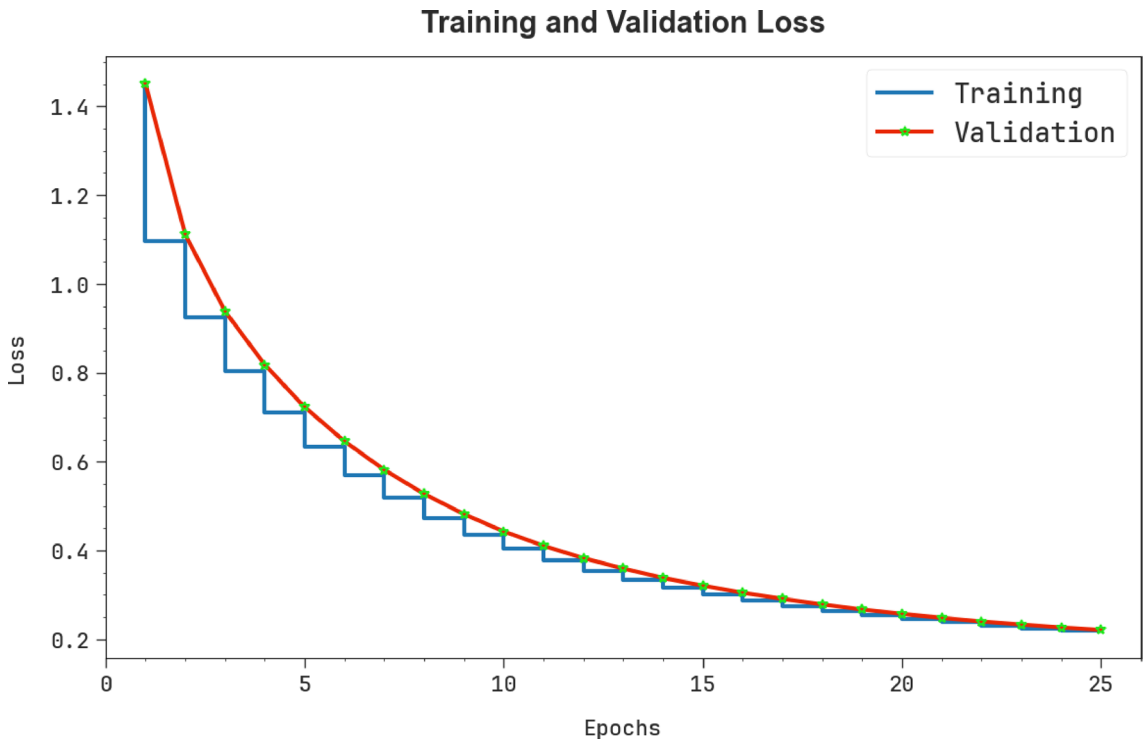

Fig. 7. Loss curve of FOFSDL-SCD method.

Technique	$Accu_y$	$Prec_n$	$Recal_l$	$F1_{score}$
LSTM-KPCA	98.00	91.00	84.00	87.00
Stacked Unsupervised FL	98.00	88.00	91.00	90.00
FL-MA	99.20	95.10	95.50	96.02
Generic CNN	92.10	90.20	90.10	90.79
Xception	92.10	89.76	92.10	90.22
VGG16	97.69	89.79	89.79	87.41
InceptionResnetV2	88.37	84.35	88.37	85.22
FOFSDL-SCD	99.38	96.27	96.26	96.27

Table 3. Comparative analysis of FOFSDL-SCD method with existing techniques.

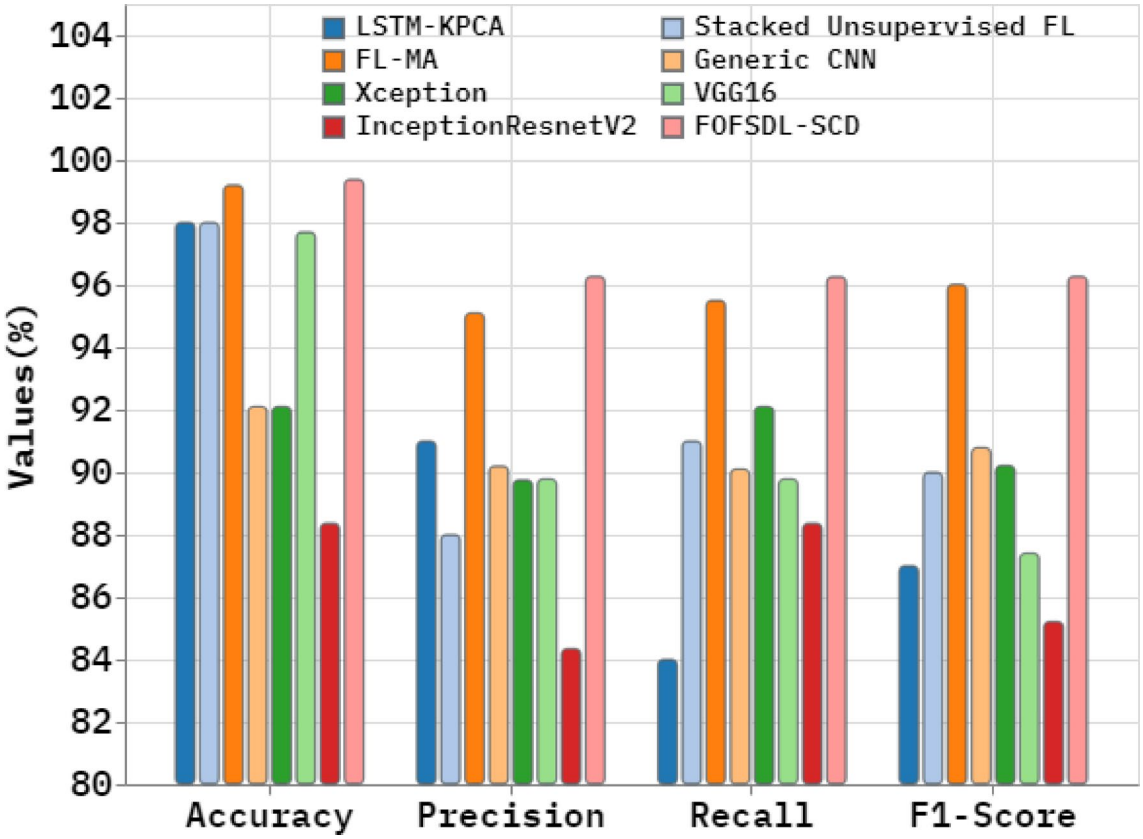

Fig. 8. Comparative analysis of FOFSDL-SCD method with existing techniques.

the superior classification capability and robustness of the FOFSDL-SCD method in accurately detecting cyber threats within complex IoT environments.

1. **Initialization**:
2. Randomly initialize the positions of all dung beetles in the search space.
3. Each beetle represents a candidate solution x_i.
4. **Fitness Evaluation**:
5. Compute the fitness value for each beetle using the objective function.
6. **Rolling Behavior**:
7. Simulate forward rolling using:

$$x_i(t+1) = x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x$$

Where $\Delta x = |x_i(t) - X^{\omega}|$ and X^{ω} is a poor position to avoid.

8. **Dancing Behavior** (if obstacle encountered):
9. Change direction based on:

$$x_i(t+1) = x_i(t) + tan(\theta)|x_i(t) - x_i(t-1)|$$

10. **Breeding Behavior**:
11. Detect safe zones near local best and generate new positions.
12. **Foraging Behavior**:
13. Search globally for optimal areas for better fitness.
14. **Stealing Behavior**:
15. Move towards elite solutions by replicating better beetles' positions.
16. **Update Positions**:
17. Apply the corresponding update rule for each beetle.
18. **Loop Until Convergence**:
19. Repeat steps 2–8 until a termination criterion is met (e.g., max iterations).
20. **Output**:
21. Return the best solution found during the process.

Algorithm 1. DBO technique.

Table 4; Fig. 9 specify the computational time (CT) analysis of the FOFSDL-SCD approach with existing models. The FOFSDL-SCD approach demonstrates a competitive CT of 7.92 s, significantly faster than prevalent models such as VGG16 and InceptionResnetV2, which require 24.15 s and 28.76 s, respectively. Other models like LSTM-KPCA and Generic CNN have computation times of 16.75 s and 12.55 s, while FL-MA operates in 10.90 s. This reduced CT highlights the efficiency of the FOFSDL-SCD model, making it appropriate for real-time cyberattack detection in IoT environments where both speed and high accuracy of 98.67% are critical

Technique	CT (sec)
LSTM-KPCA	16.75
Stacked Unsupervised FL	12.47
FL-MA	10.90
Generic CNN	12.55
Xception	15.26
VGG16	24.15
InceptionResnetV2	28.76
FOFSDL-SCD	7.92

Table 4. CT evaluation of FOFSDL-SCD approach with existing models.

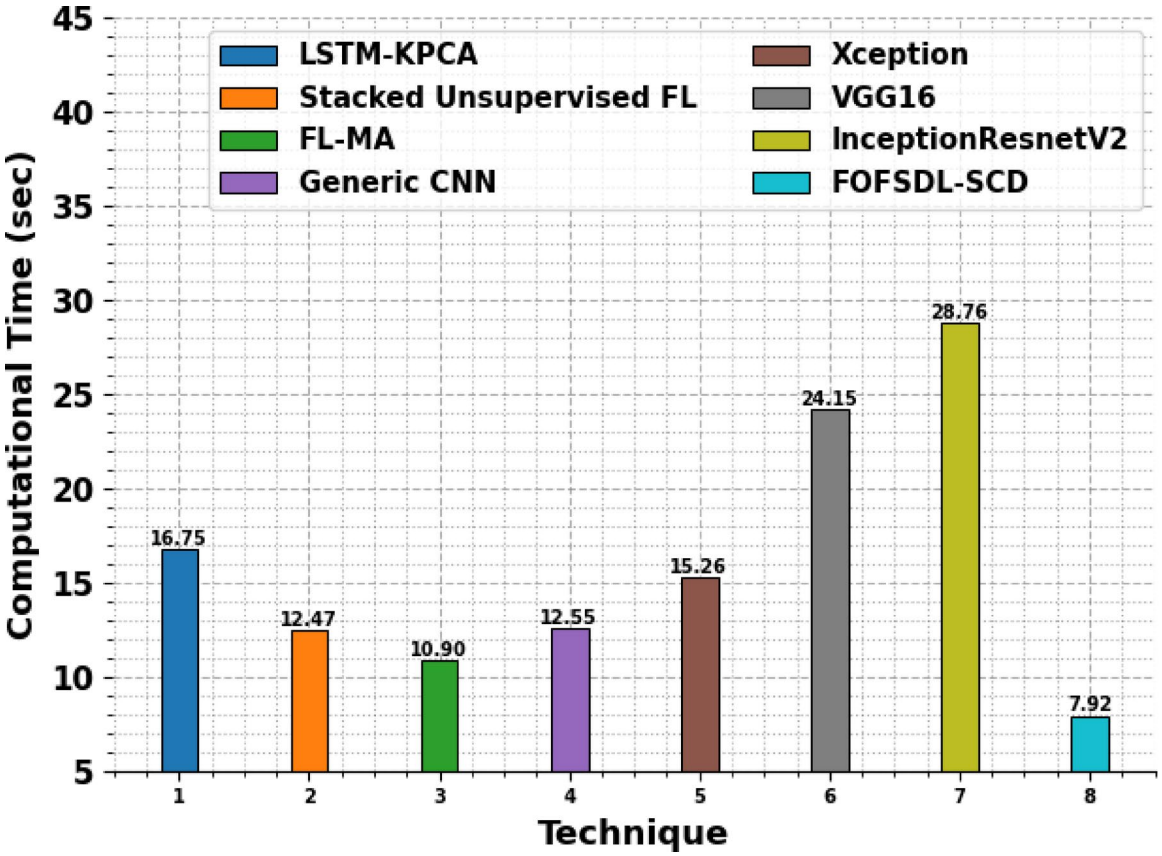

Fig. 9. CT evaluation of FOFSDL-SCD approach with existing models.

for timely responses and adequate security. The balance between minimal latency and high precision ensures effective deployment of the model in resource-constrained, security-sensitive applications.

Conclusion

This paper presents a FOFSDL-SCD model. This paper aims to analyze cybersecurity-driven approaches for enhancing IoT networks' resilience and threat detection capabilities using advanced techniques. Initially, the data pre-processing stage uses the min-max normalization method to transform the input data into a beneficial system. Furthermore, the FOFSDL-SCD model utilizes the FOA for feature selection to select the most significant features from the dataset. The TCN model is employed for the classification process. Finally, the DBO-based hyperparameter selection method is performed to improve the classification outcomes of the TCN model. The performance validation of the FOFSDL-SCD approach is examined under the Edge-IIoT dataset. The comparison study of the FOFSDL-SCD approach demonstrated a superior accuracy value of 99.38% over existing models. The limitations of the FOFSDL-SCD approach comprise limited cross-layer analysis of attack patterns and insufficient consideration of adaptive threat landscapes. Many studies lack evaluation on heterogeneous IoT datasets, mitigating generalizability. Real-world issues like latency and energy use are underexplored, and lack of decision explainability reduces user trust. Further work may concentrate on multi-source data fusion to improve

detection accuracy. Integrating real-time contextual awareness can enhance system responsiveness. Enhanced collaboration across distributed nodes can also strengthen resilience against complex attack vectors.

Data availability

The data supporting 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, reference number[33].

Received: 7 May 2025; Accepted: 22 July 2025
Published online: 06 August 2025

References

1. Alrayes, F. S., Amin, S. U. & Hakami, N. An adaptive framework for intrusion detection in IoT security using MAML (Model-Agnostic Meta-Learning). *Sensors* (*Basel, Switzerland*), **25**(8), 2487 (2025).
2. Driss, M., Berriche, L., Atitallah, S. B. & Rekik, S. Steganography in IoT: A comprehensive survey on approaches, challenges, and future directions. *IEEE Access* (2025).
3. Rehman, A. et al. A novel hybrid fuzzy logic and federated learning framework for enhancing cybersecurity and fraud detection in IoT-enabled metaverse transactions. *Egyptian Inform. J.* **30**, 100668 (2025).
4. Sharma, A., Rani, S. & Driss, M. Hybrid evolutionary machine learning model for advanced intrusion detection architecture for cyber threat identification. *PloS One*. **19** (9), e0308206 (2024).
5. Tanveer, M., Chelloug, S. A., Alabdulhafith, M. & Abd El-Latif, A. A. Lightweight authentication protocol for connected medical IoT through privacy-preserving access. *Egyptian Inform. J.* **26**, 100474 (2024).
6. Quincozes, V. E. et al. A survey on IoT application layer protocols, security challenges, and the role of explainable AI in IoT (XAIoT). *Int. J. Inf. Secur.* **23** (3), 1975–2002 (2024).
7. Alghamdi, M. I. An investigation into the effect of cybersecurity on attack prevention strategies. *J. Cybersecur. Inform. Manage.* **3** (2), 53–60 (2020).
8. Tariq, U., Ahmed, I., Bashir, A. K. & Shaukat, K. A critical cybersecurity analysis and future research directions for the internet of things: A comprehensive review. *Sensors* **23**(8), 4117 (2023).
9. Lone, A. N., Mustajab, S. & Alam, M. A comprehensive study on cybersecurity challenges and opportunities in the IoT world. *Secur. Priv.* **6** (6), e318 (2023).
10. Maghrabi, L. A. Complex proportional assessment based neutrosophic approach for ransomware detection in cybersecurity ioT system. *International J. Neutrosophic Sci. (IJNS)* **25**(2), (2025).
11. Ragab, M. et al. Artificial intelligence driven cyberattack detection system using integration of deep belief network with Convolution neural network on industrial IoT. *Alexandria Eng. J.* **110**, 438–450 (2025).
12. Ghosh, S. Network traffic analysis based on cybersecurity intrusion detection through an effective automated separate guided attention federated graph neural network. *Appl. Soft Comput.* **169**, 112603 (2025).
13. Knieps, G. Internet of things, critical infrastructures, and the governance of cybersecurity in 5G network slicing. *Telecomm. Policy*. **48** (10), p102867 (2024).
14. Inuwa, M. M. & Das, R. A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. *Internet of Things* **26**, 101162 (2024).
15. Yang, P. Electric vehicle based smart cloud model cyber security analysis using fuzzy machine learning with blockchain technique. *Comput. Electr. Eng.* **115**, 109111 (2024).
16. 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).
17. Manickam, P. et al. Billiard based optimization with deep learning driven anomaly detection in internet of things assisted sustainable smart cities. *Alexandria Eng. J.* **83**, 102–112 (2023).
18. Aboalela, R. et al. Harnessing feature pruning with optimal deep learning-based distributed denial of service cyberattack detection on IoT environment. *Alexandria Eng. J.* **120**, 584–597 (2025).
19. Ghadi, Y. Y. et al. Integration of federated learning with IoT for smart cities applications, challenges, and solutions. *PeerJ Comput. Sci.* **9**, e1657 (2023).
20. Alrayes, F. S. et al. Leveraging sparrow search optimization with deep learning-based cybersecurity detection in industrial internet of things environment. *Alexandria Eng. J.* **121**, 128–137 (2025).
21. Ghadi, Y. Y. et al. Security risk models against attacks in smart grid using big data and artificial intelligence. *PeerJ Comput. Sci.* **10**, e1840 (2024).
22. Dhanvijay, D. M., Dhanvijay, M. M. & Kamble, V. H. Cyber intrusion detection using ensemble of deep learning with prediction scoring based optimized feature sets for IOT networks. *Cyber Security Appl.* **3**, 100088 (2025).
23. Khan, S. et al. Antenna systems for IoT applications: A review. *Discover Sustainability* **5**(1), 412 (2024).
24. Alkhalifa, A. K. et al. Hybrid Dung beetle optimization based dimensionality reduction with deep learning based cybersecurity solution on IoT environment. *Alexandria Eng. J.* **111**, 148–159 (2025).
25. Mazhar, T., Shahzad, T., Rehman, A. U. & Hamam, H. Integration of smart grid with Industry 5.0: Applications, challenges and solutions. *Measurement: Energy.* 100031 (2024).
26. Alkahtani, H. K. et al. Leveraging ensemble learning with metaheuristic optimization algorithms for an intelligent cyberattack defense framework in an IoT environment. *Alexandria Eng. J.* **129**, 103–116 (2025).
27. Mazhar, T. et al. Analysis of IoT security challenges and its solutions using artificial intelligence. *Brain Sciences* **13**(4), 683 (2023).
28. Adeke, J. M., Liu, G., Amoah, L. & Nwali, O. J. Investigating the impact of feature selection on adversarial transferability in intrusion detection system. *Computers & Security* **151**, 104327 (2025).
29. Liu, Y., Gu, J. & Qi, X. X. A bidirectional gated recurrent unit and Temporal convolutional network with A Self-Attention mechanism to improve traffic flow prediction performance. *IEEE Access* (2025).
30. Jumaah, M. A., Ali, Y. H. & Rashid, T. A. Efficient Q-learning hyperparameter tuning using FOX optimization algorithm. *Results Eng.* 104341 (2025).
31. Wang, Y. & Chen, P. Network traffic prediction based on transformer and Temporal convolutional network. *PloS One*. **20** (4), e0320368 (2025).
32. Tu, K. & Cheng, J. Enhanced Dung beetle optimization algorithm and its application in 3D UAV path planning. *Electron. Res. Archive*. **33** (4), 2618–2667 (2025).
33. https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot
34. Bukhari, S. M. S. et al. Enhancing cybersecurity in Edge IIoT networks: An asynchronous federated learning approach with a deep hybrid detection model. *Internet of Things* **27**, 101252 (2024).
35. Latif, S., Boulila, W., Koubaa, A., Zou, Z. & Ahmad, J. An optimized intrusion detection framework using deep transfer learning and genetic algorithm. *J. Netw. Comput. Appl.* **221**, 103784 (2024).

Acknowledgements
The author would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication and fortheir support.

Author contributions
All contributions done by Dr. Mimouna Abdullah Alkhonaini.

Declarations

Competing interests
The authors declare no competing interests.

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