

Original Article

Two-Stage Fault Management Framework for IoT-based Wireless Sensor Network through Resilient and Energy-Efficient Approach

Suman Murahari¹, Rekha Redamalla²

^{1,2}Department of Computer Science and Informatics, Mahatma Gandhi University, Nalgonda, Telangana, India.

¹Corresponding Author: murahari.suman@gmail.com

Received: 03 April 2026

Revised: 08 September 2026

Accepted: 18 September 2026

Published: 27 October 2026

Abstract - The Internet of Things (IoT) is built on the foundation of Wireless Sensor Networks (WSNs), which is the technology that provides a smooth connection and real-time data transmission in smart cities and industrial settings. Nevertheless, IoT-based WSNs are prone to several challenges due to hardware, software, and time-related faults, which affect stability and reliability. In the research, a two-step fault management system is proposed to improve fault detection, classification, and resilience in IoT-based WSNs. At the initial step, the identification of faulty nodes in heterogeneous environments is done with a Siamese-Residual Attention Convolutional Neural Network - Bidirectional Long Short-Term Memory (SRA-CNN-BiLSTM) model. Next, the second step used a Self-Learning Residual Shrinkage Network (SLRSN) to categorize the types of faults effectively. The optimization of the model parameters is performed using a new Clan Cooperative Crested Porcupine-based Optimizer (3CPO) that guarantees high accuracy and convergence rate. Python simulations indicate a 0.98% increase in the accuracy of fault detection and significant energy savings. The framework is therefore a reliable and scalable system of fault management of IoT-enabled WSNs.

Keywords - Internet of Things, Wireless Sensor Network, Fault management, Fault detection, Fault classification, Resilient networks, Energy-efficient recovery, Fault tolerance, Energy-efficient approach.

1. Introduction

The Electrical Secondary Distribution Networks (ESDN) are very complex with high user density, making detection of defects and faults very challenging. In many developing countries, faults in ESDN have been reported mainly by customers and through visual inspection by utility personnel [1]. In such an IoT-driven system, intelligent sensor nodes are employed to monitor and transmit a wide variety of physical parameters to a Base Station (BS) through WSNs [2]. The BS receives this information via the sensor location information and further processes that data, whereas IoT-based systems are meant to entail advanced and intelligent WSNs for real-time decision making [3], and it refers to such a network as an IoT-enabled WSN, and sensor nodes function as multi-agents [4]. Applications of IoT technologies is found in various areas, including Industrial IoT, precision agriculture, smart home and office automation, and healthcare [5].

Sensor modules in IoT-based WSNs are highly vulnerable to failures due to any sort of battery exhaustion, environmental damage, or hardware malfunction [6], and these faulty sensor nodes deteriorate the efficiency of a network [7].

The existing fault detection mechanisms suffer from low accuracy and high false-alarm rates [8]. Moreover, these methods tend to be very energy-hungry, hastening the depletion of the sensor nodes and thereby shortening lifespan [9]. To counter these problems, the research at hand proposes an intelligent fault detection technique that improves accuracy while keeping energy consumption to a minimum [10]. Faults in such IoT-enabled WSNs are essentially determined by hard faults that take place whenever critical components fail the transmitter, a certain receiver, or the battery [11]. Soft faults exist when sensor nodes are simply giving faulty readings; also, the conventional detection mechanisms are now unable to truly learn accurately from analyzing heterogeneously present huge time series data coming in from multiple sensors [12]. The optimized deep-learning fault model introduced in this contribution greatly contributes to increasing the operational efficiencies within faulty environments [13]; however, as state of an art technology, Deep Learning (DL) is getting in the limelight due to handling even large and heterogeneous datasets better than conventional Machine Learning (ML) techniques [14]. Other IoT applications that adopt DL as a key player include device identification, drone detection, and cloud service optimization [15]. The LSTM



networks come to the forefront of the most popular DL models for different IoT applications that possess heavy data loads generated by sensor nodes [16]. However, LSTM-based models depend significantly on the selection of optimal hyperparameters [17], grid search and random search have traditionally been assigned to hyperparameter search, but these processes are inefficient and resource-intensive at times [18]. Against this backdrop, metaheuristic algorithms have come into prominence with new relevance, whereby hyperparameter tuning is used as an optimizing resource for improving accuracy [19].

The introduction of a practically intelligent fault detection framework aims to introduce minimal computational overhead for the sensor nodes, therefore dramatically enhancing the overall performance of the network with IoT-enabled WSNs [20]. The current two-stage fault management methods mostly depend on the use of static DL models, which are typically constrained by slow convergence, large false-alarm rates, and ineffective parameter optimization.

The originality of the research is the combination of an SRA-CNN-BiLSTM architecture with SLRSN, which is trained using the 3CPO. This combination is the only one that makes adaptive threshold learning, real-time resilience, and energy-efficient optimization possible, which leads to much higher detection accuracy and fewer false alarms than traditional DL-based fault management systems.

The research gap addressed in this work is the absence of an integrated, energy-aware two-stage fault management mechanism that jointly performs robust faulty-node identification, adaptive fault-type classification, and optimized model tuning for heterogeneous IoT-based WSNs.

To address this gap, the proposed framework combines a Siamese-Residual Attention Convolutional Neural Network-Bidirectional Long Short-Term Memory (SRA-CNN-BiLSTM) model for faulty-node identification with a Self-Learning Residual Shrinkage Network (SLRSN) for adaptive fault classification. A Clan Cooperative Crested Porcupine-based Optimizer (3CPO) is further used to tune model parameters and improve convergence.

The novelty lies in this coordinated combination: SRA-based similarity and residual attention capture discriminative spatial-temporal fault characteristics, SLRSN suppresses irrelevant/noisy features through learned shrinkage thresholds, and 3CPO balances exploration and exploitation during optimization.

The resulting design targets high detection accuracy and low false-alarm rates while reducing energy consumption and preserving network lifetime, thereby extending beyond existing approaches that optimize only one of these objectives.

1.1. Literature Review

Recent research works have previously existed in the literature, which are based on a two-stage fault management framework for IoT-based WSN through a resilient and energy-efficient approach.

Khan et al. [21] presented the Federated LSTM model solution to temperature data invasions, along with real sensor-like imitated data, for introducing the state-of-the-art sensor faults, including bias, drift, spikes or jumps, odd behaviour, stuck values, and data loss. The centralized aspects of measuring effectiveness were based on accuracy, precision, and sensitivity. Padmasree and Chaithanya [22] suggested a Gradient Boosting Classifier (GBC) to replicate faults in sensor networks, targeting the improvement of the network systems in terms of scalability and longevity.

Moreover, the technique was found to be very effective in diagnosing faults. Godfrey et al. [23] suggested the novel idea of Software-Defined WSN (SDWSN) intended and create redundancy to counter unexpected disasters. The model was in a quest to minimize dislocation while maximizing efficiency in overall performance due to energy efficiency and quicker adaptations to the dynamism of assets and voice transmissions through enhancements in packet delivery rates and reduction of latencies for data transmission.

Gupta et al. [24] presented a Region-based Hierarchical Clustering for Efficient Routing (RHCER) scheme, which was devised as an approach to enhance IoT applications and the collection by counteracting sensor faults. Results indicate that energy consumption within sensor nodes was reduced, thereby improving the packet drop ratio.

Heidari et al. [25] presented an Multi-Criteria Decision-Making (MCDM) approach to support not only fault detection but also provide suggestions for improving system performance. Fault management techniques were developed to enhance performance by reducing damage risk and helping developers of WSN to make informed decisions. Hu et al. [26] suggested a Deep Reinforcement Learning (DRL) method for real-time fault detection in WSNs so as to keep them in stable operation.

The method updates itself dynamically according to conditions was observed experimentally that a DRL-based approach was much superior to classical ones concerning the enhancement of network life and optimization of data transmission efficiency. Sheikh and Joshi [27] presented the Improved Smart Energy-based Routing Protocol (ISERP) for network reliability through fault detection, which basically means the improvement of the lifetime of the network, the reduction of packet delay, and the prevention of potential attacks to IoT-based sensor networks. Herewith, energy consumption gets minimized, and all the nodes remain alive till the end of the phase. Aruchamy et al. [28] presented an

Energy Aware Hardware and Link Fault Diagnosis (EAHLFD) method that improved diagnosis and response efficiency. The technique guarantees a reliable diagnosis by identifying faults and selecting the optimum recovery link promptly.

The selection took three critical features into account, such as probability value, link cost and battery energy. EAHLFD achieved much improved accuracy in the fault diagnosis. Jihani et al. [29] suggested a Kalman Filter (KF) method as a means to identify sensor faults in WSNs. The advantage of this method was simplicity in implementation and computational load in return for better reliability in the application.

Dogra et al. [30] suggested a Region-based Energy Efficient Routing Protocol (REERP) tailored for WSNs of the IoT. It enabled new node integration into existing clusters and selection of a new cluster. Consequently, network life and reduced energy consumption were ensured.

1.2. Problems Identified

The literature indicates four interconnected limitations in current IoT-based WSN fault-management methods. First, methods that rely on conventional machine learning or a single recurrent architecture have difficulty representing heterogeneous spatial and temporal fault signatures. Second, fault detection and fault-type classification are often treated as isolated tasks, which limits end-to-end diagnostic consistency. Third, fixed thresholds or weak noise-suppression mechanisms increase false alarms when sensor behaviour and fault probability vary. Fourth, high computational and communication costs accelerate battery depletion, particularly as network size increases.

Accordingly, the central problem is to design a fault-management framework that accurately identify faulty nodes, classify multiple fault categories, adapts to changing sensor distributions, and preserves network energy at the same time. The problem is therefore formulated as a joint accuracy-resilience-energy challenge rather than as an isolated classification task.

1.3. Motivation

The motivation for the proposed framework arises from the trade-off observed across existing studies. Fault-detection models such as LSTM-, classifier-, Kalman-filter-, and reinforcement-learning-based approaches to improve diagnostic performance, whereas routing and clustering methods reduce energy consumption and prolong network lifetime.

However, improving one aspect does not necessarily guarantee reliable operation under changing fault probabilities and heterogeneous node conditions. A practical IoT-based WSN requires a unified mechanism in which feature learning,

fault classification, optimization, and energy-aware operation support one another. For this reason, the proposed framework integrates SRA-CNN-BiLSTM, SLRSN, and 3CPO rather than relying on a single detector or a conventional optimizer.

The SRA-CNN-BiLSTM learns discriminative spatial-temporal representations; the SLRSN uses adaptive shrinkage to suppress irrelevant features and retain fault-related information; and the clan-cooperative search strategy of 3CPO is intended to improve convergence and reduce the risk of premature optimization.

This combination constitutes the principal methodological novelty and is designed to achieve a better balance among fault-detection accuracy, false-alarm reduction, computational efficiency, energy consumption, and network lifetime than existing single-stage or separately optimized fault-management approaches.

1.4. Objectives

The objectives of this research are as follows,

- To improve reliability and resilience, a new two-stage framework of the IoT-based WSNs is developed, which combines fault detection based on the SRA-CNN-BiLSTM model and fault classification based on the SLRSN model.
- To optimize model parameters, the 3CPO, a new metaheuristic optimizer, is presented to enhance the rate of convergence and the overall detection rate.

1.5. Organization

The structure of the research is given as: Section 2 presents the Explanation of the proposed framework for IoT-based WSN. Section 3 contains a description of the results analysis, and finally, Section 4 determines the conclusion.

2. Proposed Framework for IoT-based Wireless Sensor Network

The proposed framework presents a novel technical approach to design a two-stage fault management framework that integrates DL and optimization into one model of IoT-based WSNs. In contrast to current single-stage or traditional detection methods, the framework used SRA-CNN-BiLSTM to accurately identify faulty nodes and SLRSN to adaptively classify faults. Another model is a 3CPO that is optimized to efficiently optimize model parameters to guarantee faster convergence and better accuracy.

The combined effect of these mechanisms is to increase the detection accuracy, fault tolerance and energy efficiency, creating a clear novelty and technical improvement over the previous works. Figure 1 shows the multi-tiered strategy thus increases the fault detection accuracy with a lower rate of false alarms.

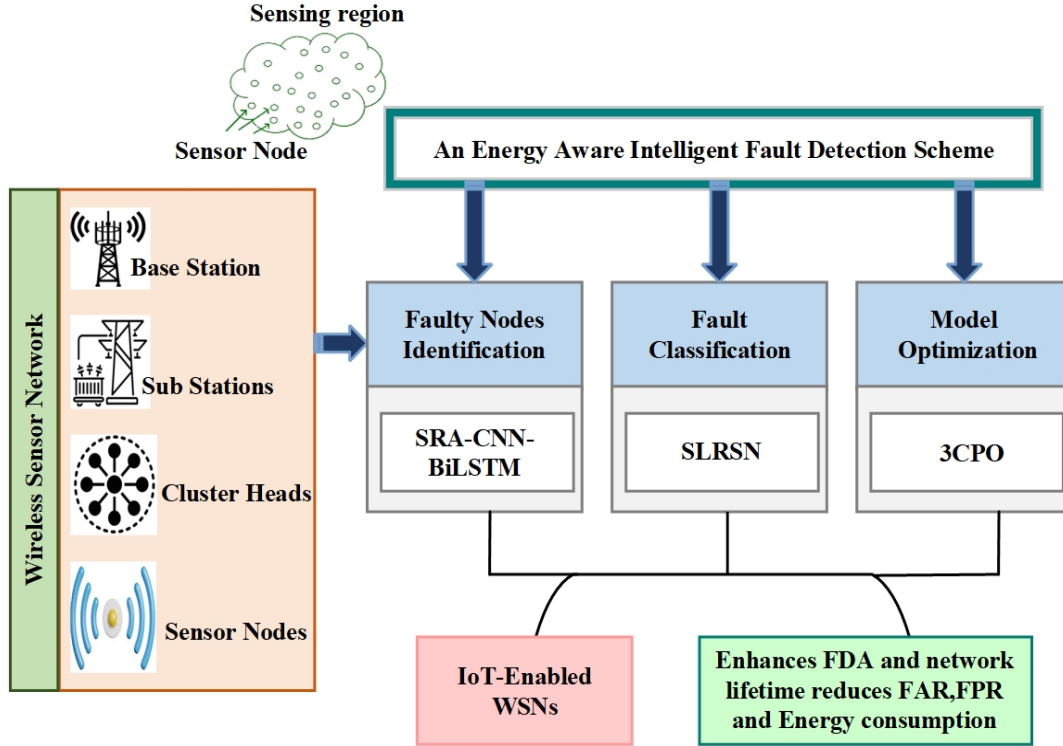


Fig. 1 Proposed framework for IoT-based WSN-based fault management

2.1. System Model

The IoT-based WSN considered in this research consists of N sensor nodes $S = \{S_1, S_2, \dots, S_N\}$, a set of cluster heads $C = \{C_1, C_2, \dots, C_k\}$, and a BS. Each node collects environmental data and transmits it either directly or via the cluster head to the BS [31].

The transmission energy E_t and reception energy E_r are modelled as per Equations (1) and (2).

$$E_t(d) = E_{elec} \times b + E_{amp} \times b \times d^n \quad (1)$$

$$E_r = E_{elec} \times b \quad (2)$$

Where b and d are the number of transmitted bits and transmission distance, E_{elec} and E_{amp} denotes energy consumed per bit and amplifier energy constant. Nodes experience hardware, software, and time-based faults that alter the sensed data by the Equation (3).

$$D_i(t) = D_{ref}(t) + \epsilon_i(t) \quad (3)$$

Where, $\epsilon_i(t)$ is the fault deviation component.

2.2. Problem Formulation

The proposed framework aims to maximize the accuracy of the fault detection and classification while minimizing

network energy consumption. The optimization objective is expressed by Equation (4).

$$\min_{W, \theta} \lambda_1 F_r + \lambda_2 F_a + \lambda_3 E_c \quad (4)$$

Subject to Equation (5):

$$E_i \leq E_{max}, D_q \geq D_{min}, R_i \leq R_{max} \quad (5)$$

Where, F_r and F_a are the fault rate and false alarm rate, E_c denotes total energy consumption, λ_i indicates the weight coefficients. Also, W and θ denote the model and optimizer parameters of the SRA-CNN-BiLSTM and SLRSN networks, respectively. This formulation defines the research problem as a multi-objective optimization problem, distinguishing it from existing models that treat fault detection and classification separately.

2.3. Fault Detection Framework

A hard fault happens when a sensor node sends information at a specific moment but does not receive any information back, or when the node receives information but is unable to transmit it. The node is totally unresponsive in some cases and does not send or receive information. A gain fault occurs when the sensed value obtained is scaled. The fault modelling phase mathematically models the sensor nodes in IoT-based WSNs to display fault behaviour. The models

assist in identifying the presence of a hard, soft, or gain-related fault in a node. The Equations (6) to (9) below demonstrate various kinds of faults and thresholds to be utilized in fault identification. The expressions are represented as follows,

$$Rg_{new} = R_{old} \cdot \alpha + \nu \quad (6)$$

Equation (6) indicates the gain fault model, where the sensed output is linearly dependent on the input with a scaling factor and a constant offset. Such scaling deviations typically occur due to calibration drift or sensor degradation, where Rg_{new} denotes the output variable, α denotes the scaling factor, ν represents the constant, R_{old} denotes the input variable.

$$R_{new} = R_{old} + \omega + \nu \quad (7)$$

Equation (7) defines the relationship between the position vector and the angular velocity ω , which assists in identifying anomalies in node movement, a common cause of mechanical faults, where, R_{new} , R_{old} denotes the position of vectors, and ω represents the angular velocity.

$$R_{new} = R_{old} \cdot \prod_{i=0}^t \phi_i \quad (8)$$

Equation (8) introduced a time-based variable that captures fault occurrences over a specific instance t , allowing the system to track temporal fault propagation, where, ϕ_i denotes a constant instance of time.

$$Rob_{new} \notin [\theta_1, \theta_2] \quad (9)$$

Equation (9) defines the threshold conditions for detecting fault values, where, θ_1 represents the lower threshold θ_2 represents the upper threshold.

2.3.1. Use of Siamese-Residual Attention with CNN-Bi LSTM (SRA-CNN-Bi LSTM)

The model, which is employed to detect faults strongly, is the SRA-CNN-BiLSTM model, which combines temporal and spatial features to differentiate normal and faulty sensor signals as described in the following equations. It is proposed to promote strong visual tracking by encoding instance-level information of a target in the course of time. Figure 2 shows the method adopts a residual learning structure, which is organized into two main branches are identity branch and the residual branch. The identity branch (F_t) aims to extract global class-level features to distinguish between normal and faulty node behaviors. Concurrently, the residual branch (F_r) is online learned based on the target as well as visually close distractors, led by the identity branch. This branch has an essential function to learn instance-specific variations and is dynamically updated through backpropagation. The fusion weights α and β are computed in terms of entropy of each

branch output to balance the contribution is defined by Equation (10).

$$f_{fusion} = \alpha \cdot F_t + \beta \cdot F_r, \text{ where } \alpha + \beta = 1 \quad (10)$$

where α and β are entropy-derived fusion coefficients. The entropy-based fusion ensures that the most informative branch dominates when uncertainty increases.

$$F_R = \frac{P - C_t \cdot F_t}{C_R} \quad (11)$$

Equation (11) indicates the computation of fusion weights according to the entropy of the outputs of the identity and residual branches. These weights dynamically balance between the general and instance-specific learning during feature extraction, where, C_t and C_R are denoted as the fusion weights and computed based on the information entropies of the corresponding branch output. F_R and F_t denoted as the residual and identity branch. In Figure 2 $\mathcal{X}_i$ and represent the input feature maps of the two branches of the Siamese network, which are the target and the candidate regions, respectively. Let F_t and F_r represent the output characteristics of identity and residual branches. Also, $\mathcal{W}_t$ and $\mathcal{W}_r$ denote the adaptive fusion weights of the two branches. The variable $s(\mathcal{X}_i, \mathcal{X}_j)$ measures the similarity score between the two feature embeddings, and y represents the ultimate classification output, which defines the level of fault correspondence. Equations (12)-(14) are always applied to compute the overall similarity-based fault detection response using these variables. Outputs derived by the identity and residual branches are modelled as different features that have contributions towards tracking accuracy in totality through Equations (12)-(14).

$$F_t(x, s) = \beta(x) * \beta(s) \quad (12)$$

$$F_r(x, s) = \omega^2 * \phi(x) * \phi(s) \quad (13)$$

$$L = \|\gamma(F_t(x, s)) * (y - F_r(x, s))\|_2^2 + \beta \|w\|_2^2, \quad (14)$$

Equations (12)-(14) are the calculation of the final similarity score and the fusion procedure of the identity and residual branches. In particular, in Equation (12), the fusion weights are used to compute the weighted sum of the outputs of the two branches $\mathcal{W}_t$ and $\mathcal{W}_r$. Similarity metric between the feature vectors is defined in Equation (13) and the normalized response that is used in the ultimate fault classification is defined in Equation (14). This hierarchical expression enables the model to balance both the identity-level discrimination globally and instance-level residual correction [32].

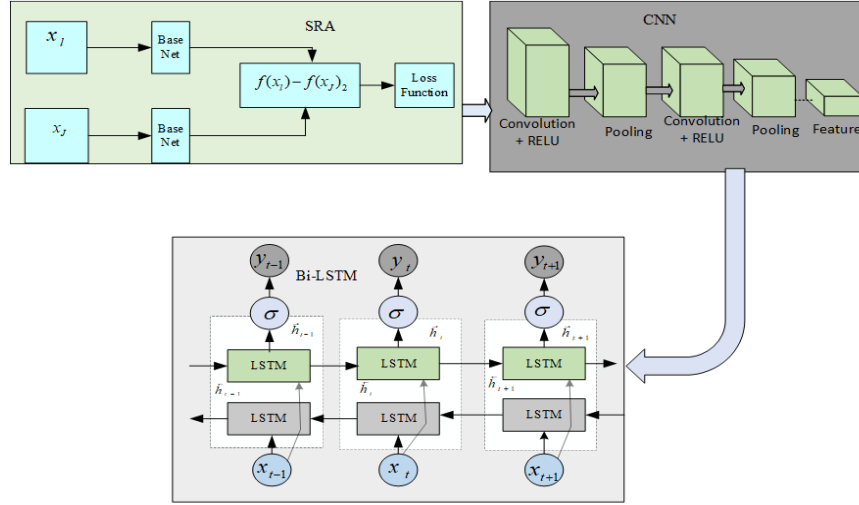


Fig. 2 Block diagram for SRA-CNN-Bi LSTM

A CNN is a feedforward neural network that is based on biological operations. The convolutional layer's purpose is to extract features by executing convolution operations to find patterns in the data. An activation function is then called upon to introduce nonlinearity, which helps the network learn more complex features. The mathematical formulation of the convolution layer is given as follows,

$$x_j^l = \sum_{i \in M_j} x_i^{l-1} \cdot k_{ij}^l + b_j^l \quad (15)$$

where x and w are the input data and kernel weight, b represents the bias term, M_i represents input data, x_i^{l-1} denotes feature vector in the $l-1$ th layer, k_{ij}^l indicates a convolution kernel vector, b_j^l signifies bias of the feature vector in the l th Layer.

$$X_m = \max_{(k-1)N+1 \leq s \leq kN} \{f_{m-1}(s)\} \quad (16)$$

Equations (15) and (16) express the convolution and pooling operations of the CNN layers. The pooling function reduces dimensionality while preserving essential patterns, where X_m represented as the pooling operation result of m th layer, N indicates pooling kernel size, and f_{m-1} denotes neuron value in $l-1$ th Layer.

$$\Pi(y) = e^y \cdot \sum_{j=1}^n e^{y_j} \quad (17)$$

Where $\Pi(y)$ represents the probability on input n represented in certain forms is defined by Equation (17). The Bidirectional LSTM (Bi-LSTM) contains a forward and backward Bi-LSTM network, which extracts the vibration signal's before and after information characteristics defined by Equations (18)-(23) captures both forward and backward temporal dependencies of the sensor data.

$$a_t = w_1 x_t + w_2 h_{t-1} \quad (18)$$

$$b_t = w_3 x_t + w_4 h_{t+1} \quad (19)$$

$$c_t = w_5 \vec{h}_t + w_6 \overleftarrow{h}_t \quad (20)$$

Where, $\vec{h}_t, \overleftarrow{h}_t$ denotes the forward and backward hidden layers of Bi-LSTM network, w_1, w_3, w_5 denotes weight of the unit function, f is the activation function. This bidirectional structure allows the model to learn from both contexts, improving fault detection accuracy.

$$i_t = \gamma(M_i[r_{n-1}, y_t] + p_i) \quad (21)$$

$$E_t = \gamma(M_f[r_{n-1}, y_t] + p_f) \quad (22)$$

$$F_t = \gamma(M_o[r_{n-1}, y_t] + p_o) \quad (23)$$

Where γ represents the activation function, M denotes weight of the corresponding gate, r_{n-1} denotes the output of the Bi LSTM unit, y_t represents input at the current moment [33]. The proposed hybrid framework also allocates diagnostic computation among sensor nodes, unlike the centralized fault detection schemes, which are highly reliant on the processing of the BS. This decentralized architecture has the benefit of minimizing latency, minimizing single-point failure, and is energy efficient and synchronized through 3CPO.

2.3.2. Implementation of SLRSN Method

The SLRSN model improves the classification of faults by reducing noise and boosting the importance of features. This expounds on the challenge of different data distributions as induced by varying gear operating conditions and presents an intelligent fault diagnosis methodology utilizing the theoretical under SLRSN methods described as shown in

Figure 3. The variable selection, which is tasked with learning a set of thresholds that are used to apply some soft-thresholding to each channel. As such, this procedure acts as an important feature is emphasized by large absolute values, while those not relevant are suppressed with small absolute values [34]. The mathematical description is given below of the feature thresholding and learning process. The expressions are given as,

$$w_m^{l+1} = \text{ReLU}(\sum_p x_p^l \cdot k_{pm}^l + b_m^l) \quad (24)$$

$$y_m^{l+2} = v(w_m^{l+1}) \quad (25)$$

Where, w_m^{l+1} represents the output vector of the $l + 1$ layer of the one-dimensional convolution, k_{pm}^l denotes weight, b_m^l denotes the bias.

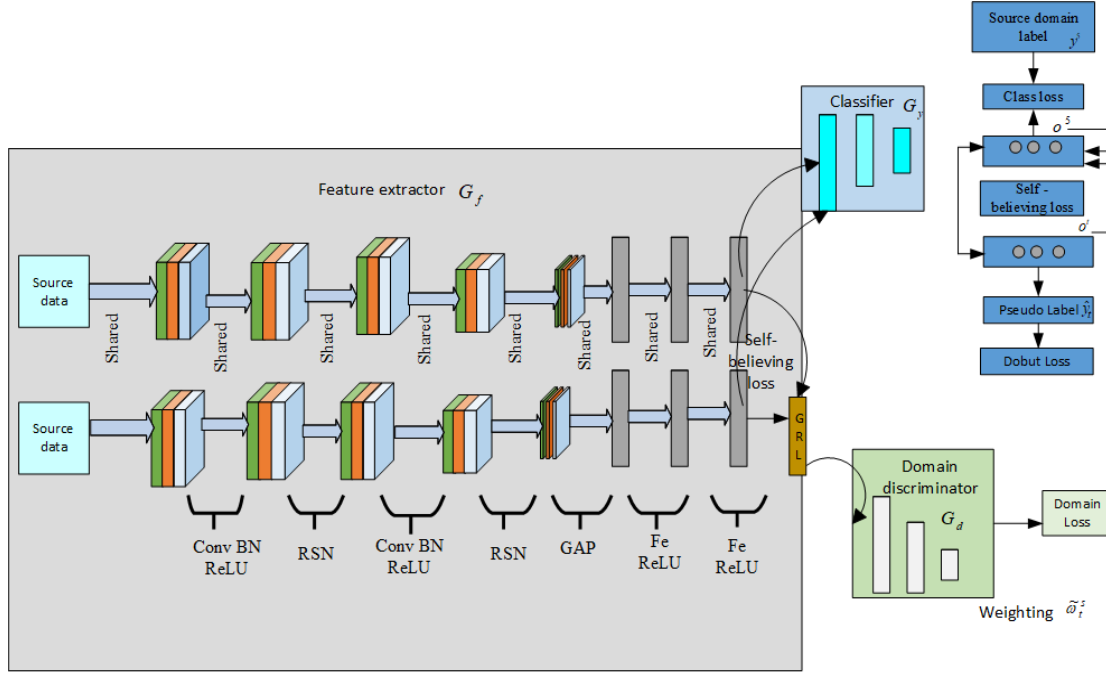


Fig. 3 SLRSN framework method

The one-dimensional convolution and SLRSN network are superimposed twice to extract deep features, the specific expressions are given as,

$$x + 1 = \text{average}(y, x) \quad (26)$$

$$y[i] = \text{ReLU}(M \cdot x_j + d) \quad (27)$$

$$y[i] = \text{softmax}(\hat{y}[i]) \quad (28)$$

Equations (24)-(28) defines how each channel in the network undergoes adaptive shrinkage based on learned thresholds, where, x_j denotes input vector, M and d are the weights of the bias, $y[i]$ denotes input sample classification, softmax denotes the final output features of the classifier. This mechanism ensures that irrelevant fault patterns are minimized, allowing the classifier to focus on key fault characteristics.

The SLRSN is trained using a hybrid objective function that combines Mean Squared Error (MSE) with an Adaptive Shrinkage Loss (ASL) as defined by Equation (29).

$$L_{\text{total}} = \lambda_1 L_{\text{MSE}} + \lambda_2 L_{\text{ASL}} \quad (29)$$

where L_{MSE} is the MSE loss that minimizes prediction error, L_{ASL} denotes the ASL loss function, which regularises soft-thresholding to avoid over-suppression of relevant features. The parameters λ_1 and λ_2 are set to balance reconstruction and sparsity. The Adam optimizer initializes early-stage learning, after which the 3CPO optimizer fine-tunes the parameters to obtain faster convergence.

2.3.3. Optimization Using 3CPO

The 3CPO is a novel metaheuristic optimization method based on the Crested Porcupine's behavioural features. It imitates four principal survival methods of the animal: vision, sound perception, and olfactory perception are two phases mentioned [35].

To optimize the SLRSN parameters as well as improve the convergence speed, the 3CPO is incorporated, and its main stages, which are initialisation, fitness evaluation, and update mechanisms, are described by the following equations.

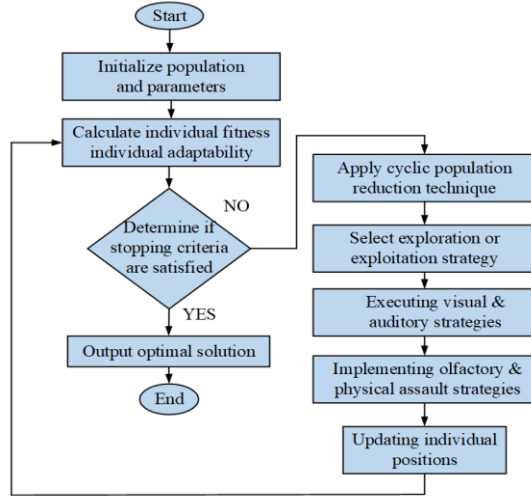


Fig. 4 Flowchart for 3CPO optimization

Figure 4 shows that the exploration phase is a global, wide search for promising areas, whereas the exploitation phase improves solutions by localized search. Similar to other metaheuristic swarm algorithms, 3CPO starts with an initial population of individuals.

2.4. Initialization

The 3CPO is a population algorithm in which clan co-operative crested porcupines are considered the search agents. All candidate solutions are 3CPOs, which are updated as part of the optimization process. Search agent position is represented by Equation (30).

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_k \\ \vdots \\ Y_{Mmax} \end{bmatrix} \begin{bmatrix} y_{1,1} \cdots y_{1,m} \cdots y_{1,p} \\ \vdots \\ y_{k,1} \cdots y_{k,m} \cdots y_{k,p} \\ \vdots \\ y_{Mmax,1} \cdots y_{Mmax,m} \cdots y_{Mmax,p} \end{bmatrix} \quad (30)$$

Equation (30) is the stage of the initialization, and the population of the porcupines (candidate solutions) is randomly spread in the search space. Every person is identified with a possible parameter set of the learning model, where Y represents the population, and P denotes the dimension size of the issue. M_{max} denotes the maximum number of individuals.

2.5. Random Generation

The random strategy identifies the behaviour of a crested porcupine as a response to predators. In this algorithm, individuals move toward the global best solution, mimicking the porcupine's survival instincts.

The strategy replicates the porcupine's ability to release a scent that deters predators. Within the algorithm, this process is represented by individuals adjusting the positions, allowing them to explore new solutions.

2.6. Fitness Function

The fitness function (F) is derived in terms of the calculation of the minimum error of the voltage and current difference between predicted and actual fault state. The fitness function that is used to compare various solutions is given in Equation (31).

$$F = \text{Min}(E) \quad (31)$$

Where E denotes the minimum current error. The fitness function is expected to evaluate the level of satisfaction of a given solution to the optimization goal. The fitness matrix is represented by the following Equation (32).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ F_{Mmax} \end{bmatrix} \begin{bmatrix} f(y_1) \\ \vdots \\ f(y_i) \\ f(y_{Mmax}) \end{bmatrix} \quad (32)$$

The fitness matrix F is the fitness of each candidate solution, and f is the objective. The solution with the lowest fitness is the fittest solution. In optimization, the fitness function evaluates all candidate solutions, and the fitness values of all the individuals of the population are obtained and stored in F .

2.7. Updation

The exploration phase of the original 3CPO algorithm replaces the position with a random update that is close to the last optimal position, which will, in effect, lead to either oversized or undersized updates. Yet, minor changes in the initial steps result in slowness of the convergence of the algorithm, and major changes in the final steps result in the search going beyond the edge of the search space and failing to find the best solution.

$$\omega = \exp \left[1 - \left(\frac{y^{1.3}}{y_{3max}} (l) \right)^{0.5} \right]^n \quad (33)$$

The adaptive position update strategy of 3CPO is explained by the Equation (33), the update factor is changing dynamically depending on the number of iterations, also the boundary conditions avoid premature convergence and give global optimization performance, where ω is used to indicate that the position of the update factor does not only adaptively decreases with the iterations, but also automatically modifies the update factor depending on whether the results of the past updates surpass the limit.

2.8. Termination

The termination processes are also updated at the same time in order to locate the positions of clan co-operative crested porcupines, and this is then followed by the fitness value. The most optimal solution is chosen based on the fitness value that is maintained.

3. Results and Discussion

The results of simulation and performance analysis concerning the proposed two-stage fault management framework regarding IoT-based WSN are presented in this section.

The analysis mainly focuses on four indicators: fault detection accuracy, false alarm rate, energy consumption, and network lifetime are interpreted with the quantitative analysis to substantiate the effectiveness, reliability, and scalability of the framework.

3.1. Experimental Setup

The experiments are conducted with Python is used to implement all the simulations with NumPy, TensorFlow, subject to the same simulation parameters for all of the models compared to facilitate fair evaluation. The experiments are run on an Intel i7@3.0 GHz, 32 GB RAM, Windows 11 operating system workstation. The sensor nodes are modelled based on IEEE802.15.4 and resemble the distributed deployment of IoT-WSN.

The IoT-WSN topology is made of 100-1000 heterogeneous nodes whose transmission range is 250 m, initial energy per node is 2J, and a packet size is 512 bytes. The fault data sets are created synthetically to model hardware, software and time-based faults, based on uniform random distributions with noise injection in order to recreate the disturbances in the real world.

The training is done with a batch size of 32, a learning rate of 0.001, and the Adam optimizer. The simulations are conducted for 20 cycles to assess the energy consumption, detection accuracy, false alarm rates, and network lifetime.

3.2. Analysis of Fault Detection Accuracy and Fault Classification Performance

Figures 5 and 6 are a compilation of the detection of the faults in the proposed framework in different conditions of faults and node configuration. The findings are always consistent that the proposed model is able to detect single and multi-faulty nodes with greater sensitivity and lower false-positive rates compared to other methods. In contrast, Sensor Node 6 showed a significantly high deviation of about 0.14 it reports under abnormal conditions for a single node fault.

The proposed system has detection accuracies of above 90% as the probability of faults rises, whereas other techniques exhibit a more pronounced decline in performance. The major deviation is observed at Node 18, that is 0.08, while the lowest deviations are seen at Nodes 2, 8, and 12. On the whole, these results validate the fact that the suggested two-stage framework provides a higher accuracy, computational efficiency, and energy consumption in heterogeneous IoT-WSNs.

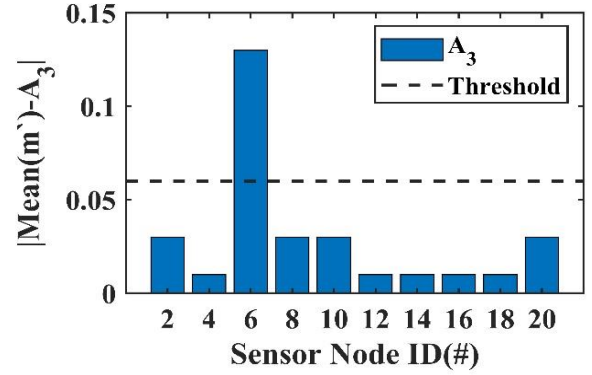
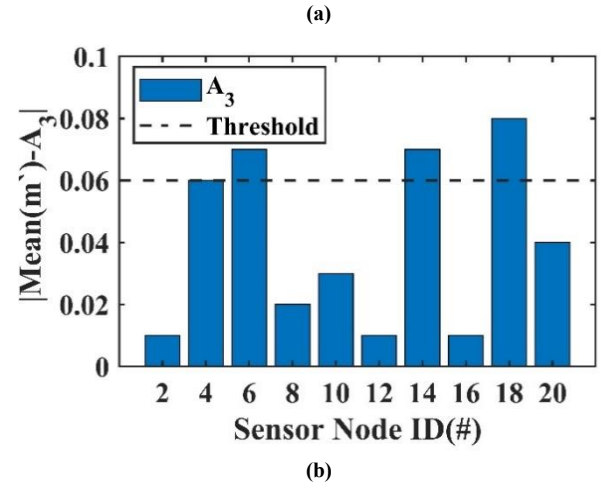


Fig. 5 Sensor node fault detection for (a) Single node fault, and (b) Four faulty nodes.

The trends in deviation, as shown in Figure 5, show that the larger the deviation magnitude, the more evidence of malfunction in the node. Single-node faults (Figure 5(a)) produce isolated peaks, and multi-node faults (Figure 5(b)) have distributed deviations between several IDs. These deviations are used to steer the adaptive thresholding in SLRSN that dynamically adjusts the fault sensitivity to reduce false alarms.

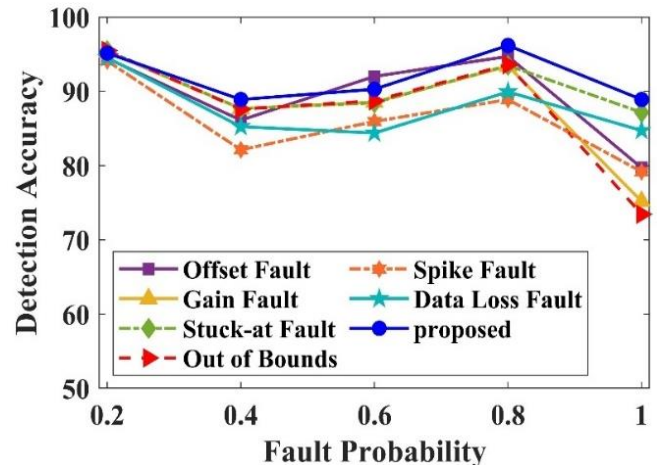


Fig. 6 CNN-based fault type detection accuracy

Figure 6 shows an increase in the number of faults, as represented in the percentage detection accuracy, that measured in terms of fault-related probabilities. When the fault probabilities are lower than 0.2, the detection accuracy is recorded at a high percentage of 90 up to even 100. However, it maintains above 85% at fault probability levels of 0.6 despite a slight decrease in detection accuracy. The accuracy varies between 80 and 95% for fault probability readings of 0.8. At the highest potential for fault probability 1.0, it hits a low point at about 70% or even lower for certain fault types. The proposed method gives the highest accuracy among all others for all faults.

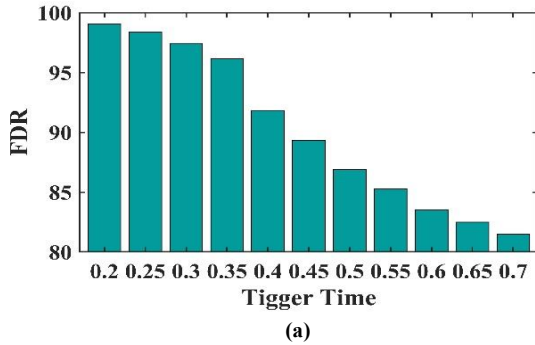
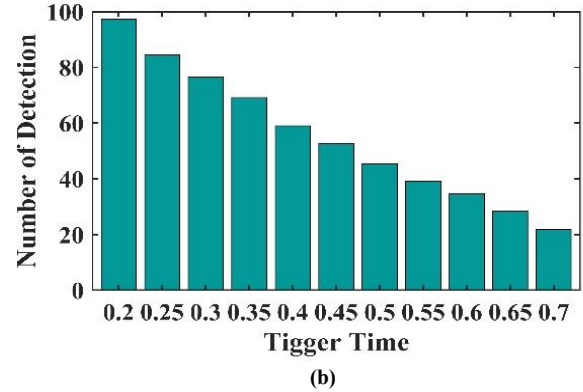
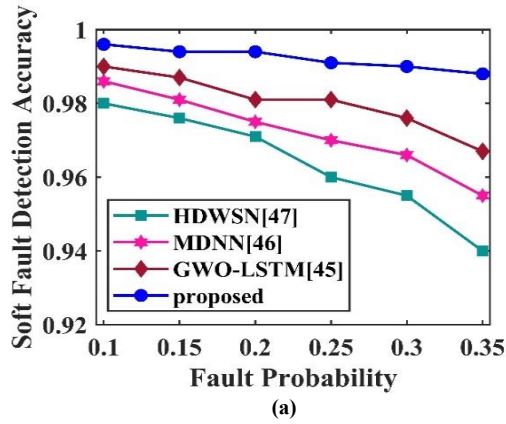
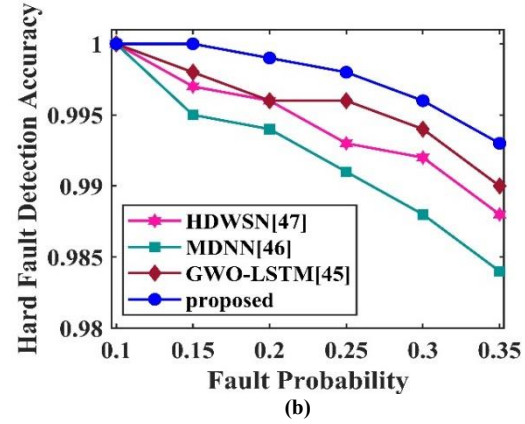
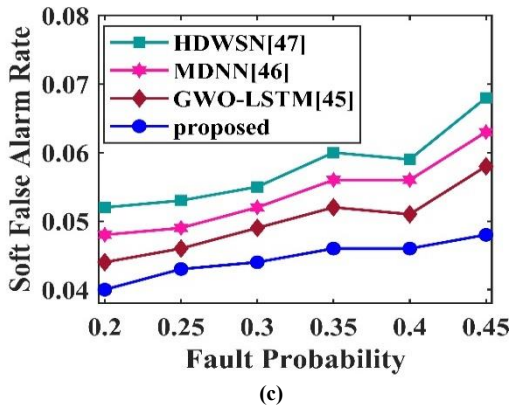
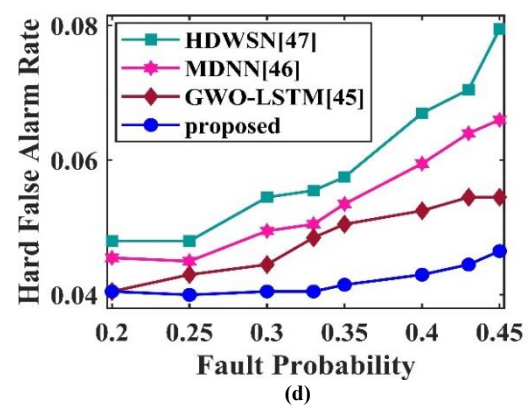


Fig. 7 Change of fault detection for, (a) Fault monitoring effectiveness, and (b) Instances of detection.

Figure 7(a) shows the trigger time from 0.2 to 0.7, which ordinarily reduces both FDR and the rate of detection. This is revealed in the graph on the left, in which FDR starts from 100% at 0.2 and becomes approximately 80% at 0.7. Similarly, Figure 7(b) shows that from almost 100 at 0.2, it drops to about 20 at 0.7 regarding the number of detections. This indicates that at a higher trigger time, there are fewer detections, plus a low false detection rate.



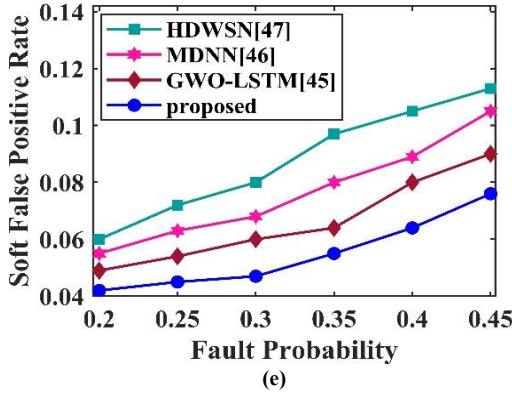
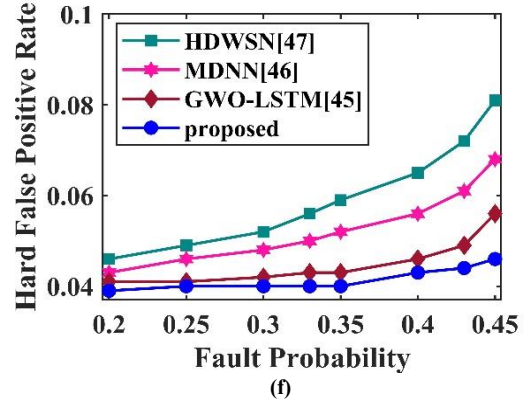


Fig. 8 Analysis of, (a) Soft fault detection accuracy, (b) Hard fault detection accuracy, (c) Soft false alarm rate, (d) Hard false alarm rate, (e) Soft false positive rate, and (f) Hard false positive rate.

Figure 8(a) shows the soft fault detection accuracy, at fault probability level 0.1, the proposed method achieves about 0.995, Grey Wolf Optimization-LSTM (GWO-LSTM) approximately 0.99, Metaheuristic Trained Neural Network (MDNN) around 0.975, and Heterogeneous fault Detection scheme for Wireless Sensor Neural Network (HDWSN) about 0.965. When the fault probability rises to 0.2, the proposed method sticks close to 0.99, whereas GWO-LSTM measures at 0.985, MDNN goes down to 0.96, and HDWSN falls to 0.945. The values at fault probability of 0.35, the highest for fault probability, continue to show above 0.98 accuracy for the proposed method; otherwise, GWO-LSTM, MDNN, and HDWSN fall short of that high threshold, and more so for HDWSN at less than 0.93. At a fault probability level of 0.1, Figure 8(b) illustrates that the Hard Fault Detection Accuracy of the proposed technique is almost 1.0, GWO-LSTM is about 0.995, MDNN is at 0.99, and is lower than HDWSN at 0.985. The proposed method continues to be the only accurate method with more than 0.985. In terms of all fault probabilities, the proposed method beats the rivals by providing maximum accuracy. The high accuracy of the proposed method is due to the joint learning power of the SRA-CNN-BiLSTM that learns both spatial and time-based dependencies of sensor data. In all probability scenarios, the proposed method remains the best solution with the least false alarm rates compared to others. Figure 8(c) shows comparisons in false alarm rates of four methods, HDWSN, MDNN, GWO-LSTM, and the proposed method against fault probability values from 0.2 to 0.45. The soft false alarm rates increase slightly for all methods with increasing fault probabilities, while the proposed method maintains a range around 0.05. Figure 8(d) shows hard false alarm rates in keeping with this growing pattern in the HDWSN method, which reaches approximately 0.08, while MDNN and GWO-LSTM rank slightly higher, still lower than the suggested approach. Figure 8(e) illustrates the behaviour of false positive rates of soft and hard fault probabilities applied for four models, HDWSN, MDNN, GWO-LSTM, and the proposed one. The HDWSN model in soft false positive rate is reaching around 0.08, while the proposed method remains in the range

of 0.11. Figure 8(f) depicts the hard false positive rate again mirrors and climbs with increasing fault probability; however, the actual numerical values are less than those in the soft case. HDWSN scores the most points again, exceeding 0.09, while the proposed model maintains a steady average of 0.11 for the entire range of fault probabilities. The trend across all the models remains unchanged, with the proposed model offering the highest efficacy as far as minimizing false positives.

3.3. Energy and Lifetime Analysis

Figure 9(a) shows a representation of energy consumption in many cycles, comparing fault detections concerning new fault detections. The cycle ranges from 2 to 20, while the consumption ranges from 0 to 0.08 *mJ*. At the beginning of the scenario, 2 cycles include approximately 0.01 *mJ* for fault detection and around 0.015 for new detection. Energy consumption for both increases as cycle numbers increase, although continuously larger than the former. By 20 cycles, fault detections in the order of 0.025 *mJ* is observed, whereas new fault detections saturate at about 0.06 *mJ* clearly showing an increasing trend. The energy usage is relatively small because of the efficient feature extraction of CNN layers and optimal parameter adjustment by the 3CPO algorithm. Figure 9 (b) shows the varying network sizes with energy consumption per joule for four different models, namely HDWSN, MDNN, GWO-LSTM, and the proposed method. On increasing the size of the network, the energy consumed by all models also increases. Significantly, at a network of 200, the energy consumed by the proposed method is nearly equal to 1J, followed by 1.3J for GWO-LSTM, 1.4J for MDNN, and 1.5J for HDWSN. The energy consumed at a network of 1000 amounts to around 2.3J for the proposed method, 3.0J for GWO-LSTM, 3.3J for MDNN, and 3.5J for HDWSN.

The proposed method is the least energy-consuming throughout all network sizes. The continuous decrease in energy consumption indicates high efficiency in computation and communication overheads of the proposed framework, thus extending the network lifetime qualitatively.

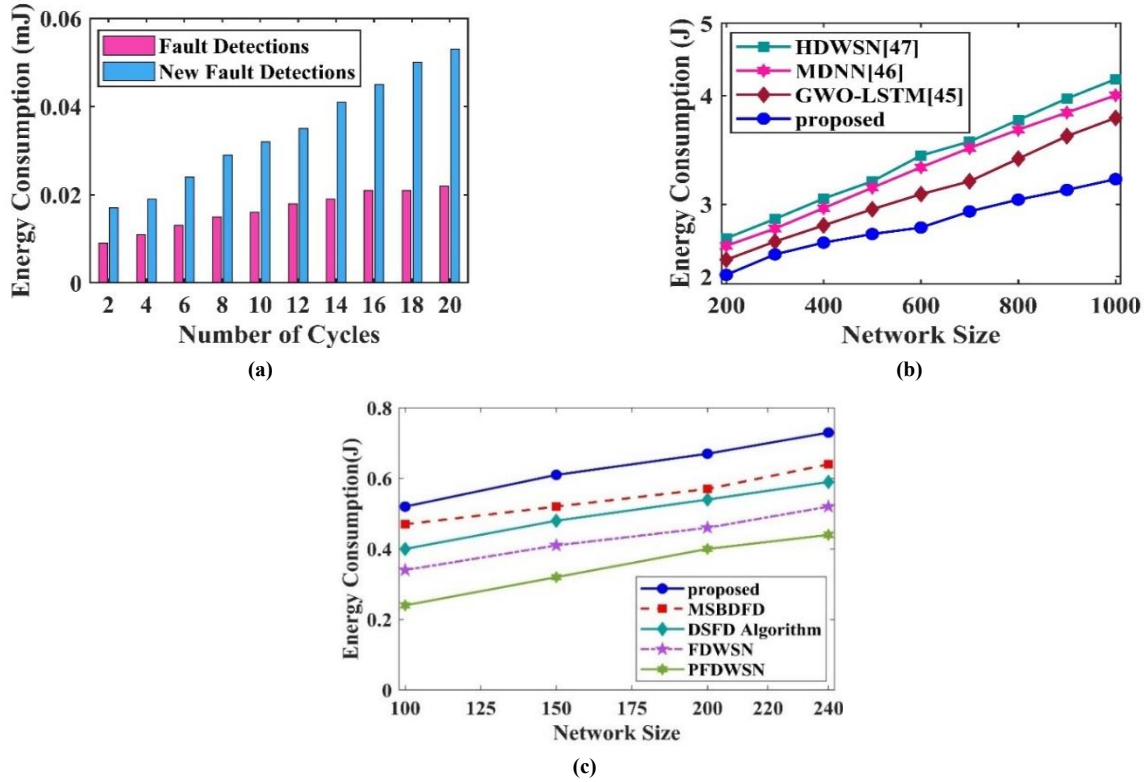


Fig. 9 Energy consumption comparison for, (a) Single node fault, (b) Network size, and (c) Fault diagnosis processes.

Figure 9 (c) shows the correlation between network size and energy consumption measured in Joules under the considerations of five different algorithms: proposed, Mobile Sink based Distributed Fault Diagnosis (MSBDFD), Distributed Soft Fault Detection (DSFD) algorithm, Fault Detection WSN (FDWSN), and Proposed Fault Detection WSN (PFDWSN).

The network size is from 100 to 240, whereas the energy consumption is approximately between 2J to 0.8J. The MSBDFD algorithm, showing rising energy consumption in the approximate range of 0.5-0.7J, follows a similar trend but consumes less energy. The DSFD Algorithm, marked by the generation of a purple star line, is started at about 0.4J and escalated to about 0.6J. The next is the FDWSN, which is slightly lagging behind the DSFD Algorithm.

Lastly, the PFDWSN extends the least energy consumption from about 0.25J to 0.45J with the increase in network size. The proposed method stands above all others during energy consumption, from approximately 0.55J at the network size of 100 to nearly 0.75J at 240. The 3CPO optimizer is the reason behind the lower energy consumption and longer network lifetime due to dynamically tuning the model parameters to achieve convergence efficiency. This reduces unnecessary computations and transmission costs, resulting in energy-conscious fault management in large-scale WSNs.

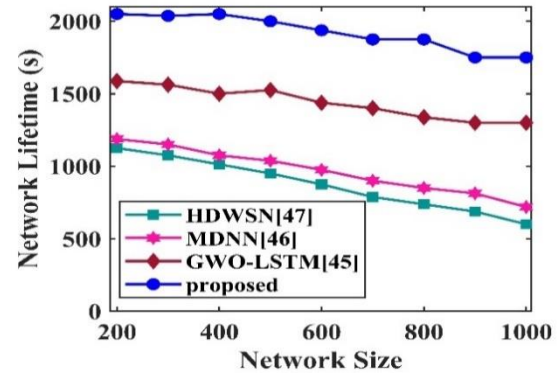


Fig. 10 Impact of node variability on network longevity

Figure 10 shows network performance of HDWSN, MDNN, GWO-LSTM, and a proposed model the network size ranging from 200 to 1000, while the network lifetime is measured in seconds, ranging from about 500 to 2500. The proposed model appears to perform better than the rest, maintaining a network lifetime above 2100 seconds throughout all network sizes. GWO-LSTM follows with a network lifetime range of 1600 seconds. The MDNN is from approximately 1200 seconds as the network size increased, while HDWSN declined from about 1100 seconds. Thus, the models show a network lifetime reduction with an increase in network size, while the proposed method shows more efficiency. A comparative analysis is therefore undertaken to validate the proposed framework's effectiveness against

HDWSN, MDNN, and GWO-LSTM. It is observed that the proposed method maintained across the board a 5 to 10% increment in fault detection accuracy, a 20 to 30% decrease in the false alarm rate, and a 25 to 35% reduction in energy consumption.

3.4. Optimizer Comparison

Figure 11(a) shows the detection accuracy rate against several fault probabilities applied on the different methods, such as Fault Diagnosis in Random Forest Classifier (FDRFC), Improved Negative Selection Algorithm (INSA), Heterogeneous Fault Diagnosis (HFD), Energy-Efficient Heterogeneous Fault Management (E²HFM), and a proposed method. Figure 11(b) depicts the fault probability in the range of 15% to 50%. The range of 80%-100% is the detection accuracy rate. Observe that the proposed method maintains the highest accuracy when the fault probability increases, to remain close to 100% HFD and INSA; however, a significantly sharper drop occurs with high reductions. The FDRFC reaches close to 80% accuracy at higher fault probabilities. Thus, the data shows the developed model illustrates a better performance level to detect very high faults at increased fault probability.

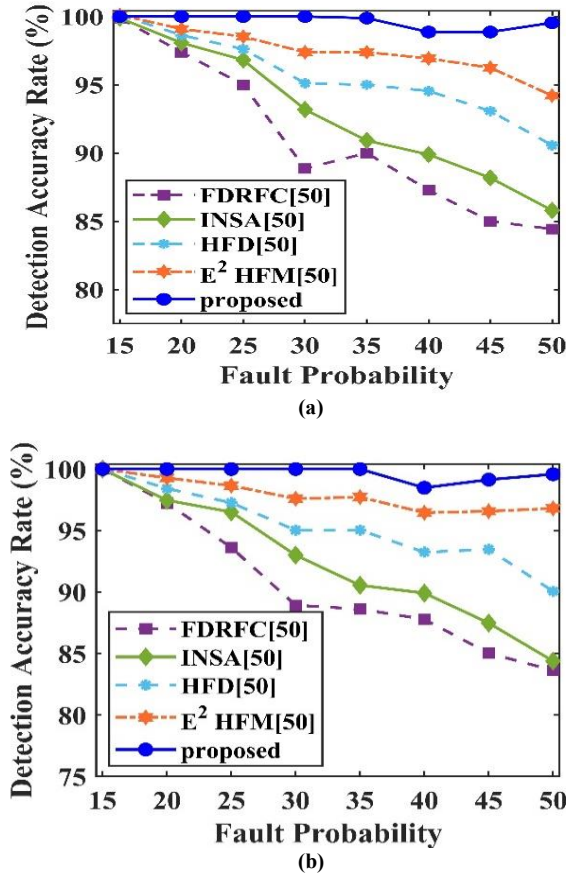


Fig. 11 Analysing the Detection Accuracy Rate (DAR) across varying fault probability rates (a) IWSN#1, and (b) IWSN#2.

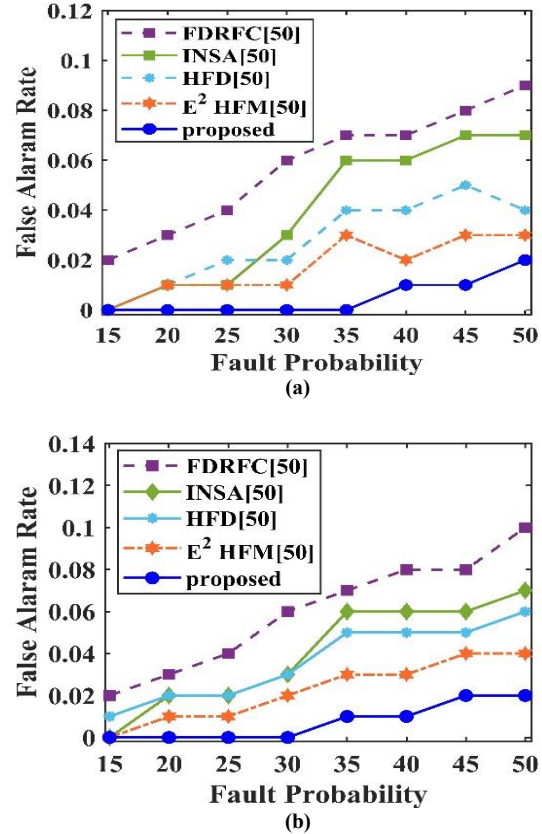


Fig. 12 Analysing False Alarm Rate (FAR) at different fault probability rates: (a) IWSN#1, and (b) IWSN#2.

Figure 12 shows five different methods, namely FDRFC, INSA, HFD, E²HFM, and the proposed method, against fault probability. The range is from 15% - 50% fault probability is from 0 to 0.1 in terms of False Alarm Rate. Out of these two graphs, Figure 12(a) and Figure 12(b), it is seen that the FDRFC method records the highest values of False Alarm Rate, which is close to 0.09 on 50% fault probability. The other two methods, INSA and HFD, show an intermediate performance, with False Alarm Rates ranging from approximately 0.03 to 0.06 with increasing fault probability. E²HFM improve slightly on False Alarm Rate as it is lower than that of HFD. This proposed method is the only one with a consistently lowest False Alarm Rate, which remains below 0.02 even at the fault probability of 50%. The low false alarm and high detection accuracy rates show that the SLRSN is a good suppressor of irrelevant features, which is represented by adaptive thresholding, which only highlights the major fault-related signals. This is selective learning that decreases overfitting and enhances the extrapolation of unobservable fault designs. The steady increase in the rate of detection and alarm reduction indicates the flexibility of the framework to different network conditions. The hierarchical learning structure of the model allows it to be reliable even when the fault probability changes, and the nodes are densely deployed.

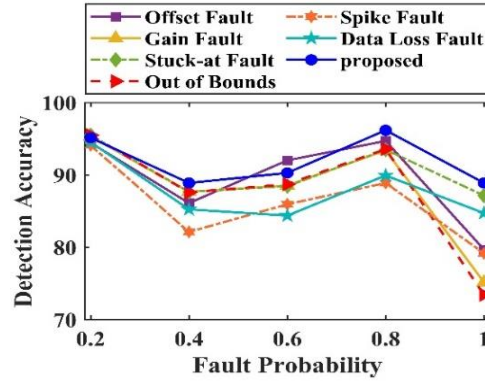
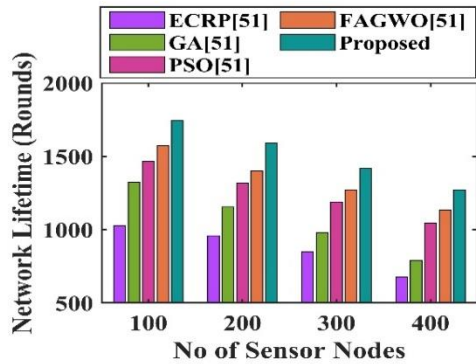


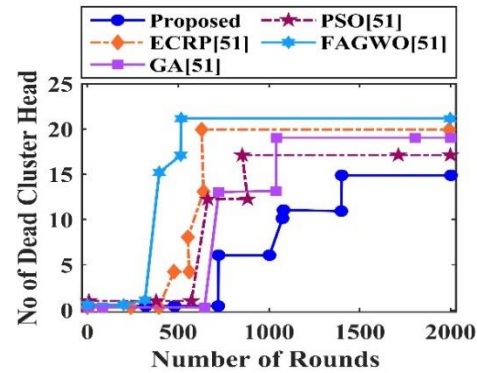
Fig. 13 Hardware failure detection accuracy

Figure 13 shows the Detection accuracy with network size on a total of five algorithms: PFDWSN, FDWSN, DSFD Algorithm, MSBDFD, and the proposed method. The range is from 100 to 240, and increments represent network size, while the range runs from 0.8 to 1.0 and represents detection accuracy. MSBDFD follows with a slightly lower accuracy of approximately 0.95 to 0.92. DSFD Algorithm and FDWSN show moderate performance with accuracy dropping from about 0.92 to 0.88. PFDWSN shows the least amount of accuracy, with 0.90 decreasing to 0.85 as the network sizes increase.

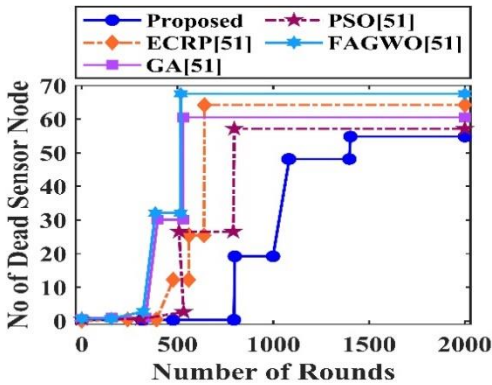
The proposed method outperforms all the others by maintaining an accuracy of greater than 0.95 as the network size increases. The network-level performance of the proposed framework is presented in Figure 14(a)-(g) using different metrics. This is because of efficient fault recovery and energy-conscious operation, which significantly extend the network lifetime. A smaller number of dead nodes and cluster heads is an indication of better stability and resilience. Also, an increased success and survival rate, lower end-to-end delay, and faster convergence prove the strength and flexibility of the proposed solution.



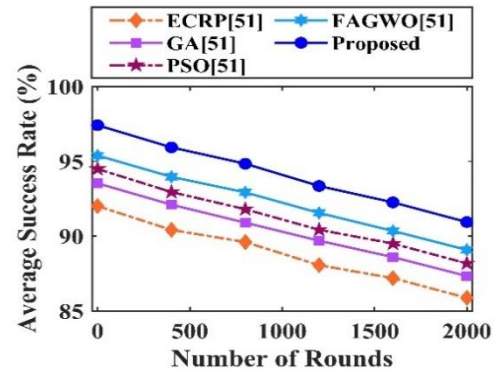
(a)



(b)



(c)



(d)

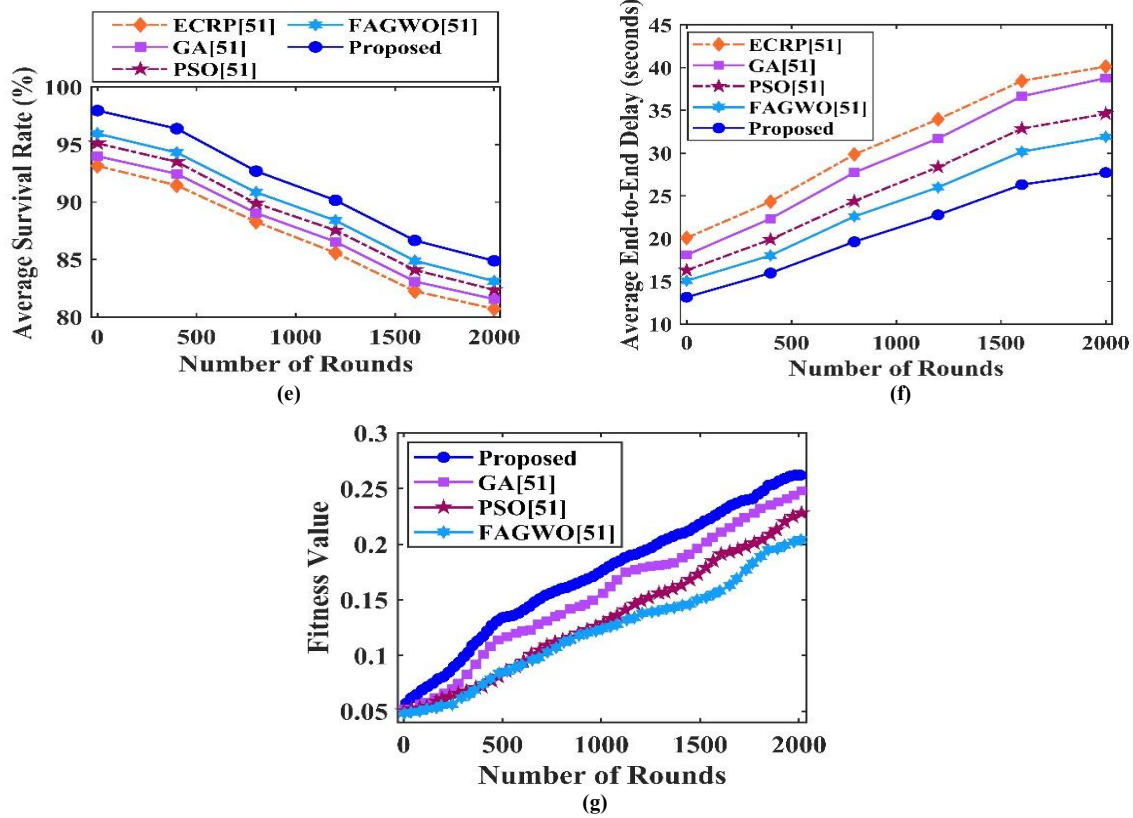
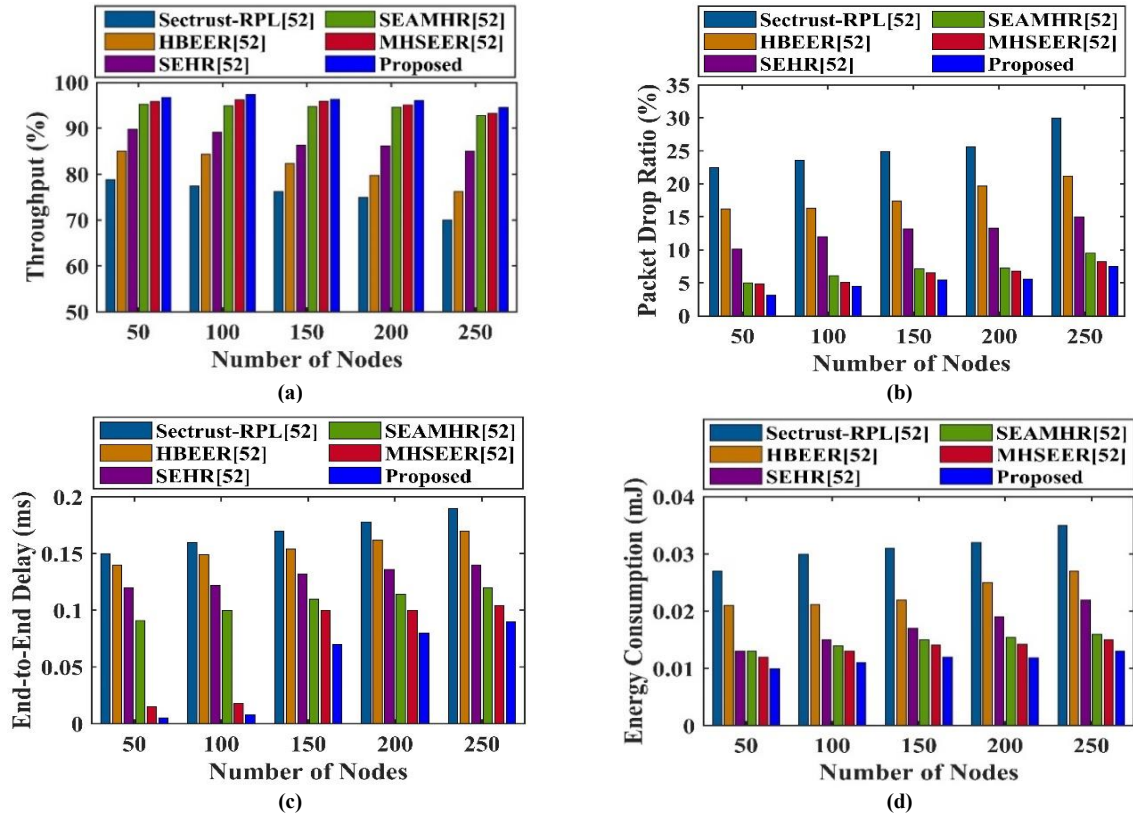
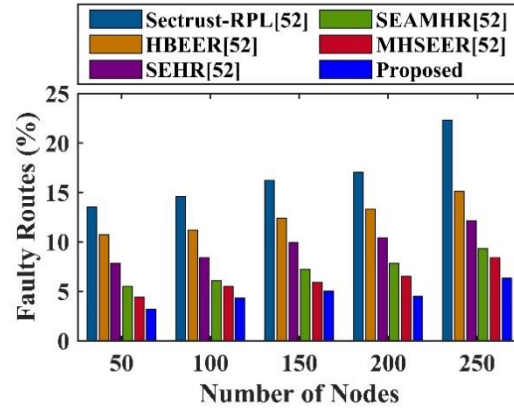


Fig. 14 Network performance comparison of, (a) Network lifetime, (b) No. of dead CH, (c) No. of dead sensor node, (d) Success rate, (e) Survival rate, (f) End-to-End delay, and (g) Convergence rate.



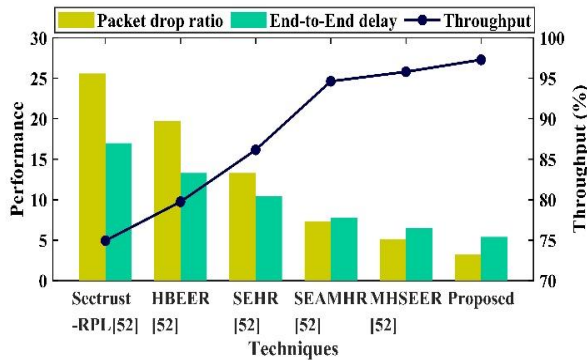


(e)

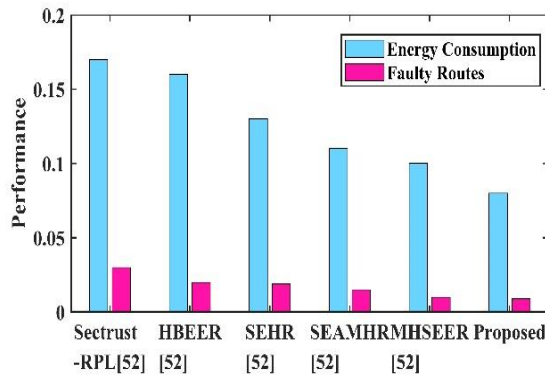
Fig. 15 Comparative analysis of communication parameters, (a) Throughput, (b) Packet drop ratio, (c) End-to-End delay, (d) Energy consumption, and (e) Faulty routes.

The performance of the developed and existing models based on communication is provided in the Figure 15(a)-(e). The framework has an increased rate of throughput and a decreased ratio of packet drop, hence data transmission is efficient. The decrease in end-to-end delay and energy consumption confirms the energy-efficient communication in the framework. Further, fewer defective paths show the reliability and fault-tolerance of the suggested model.

Figure 16 illustrates the stability and reliability of the network under fault conditions. Figure 16(a) demonstrates better packet delivery, low latency, and high throughput, which proves the effective data handling. Figure 16(b) emphasizes optimized energy consumption and reduced faulty paths, which represent even distribution of power and robust routing. They all testify to the excellence of the framework in ensuring the quality of communication and the reliability of systems.

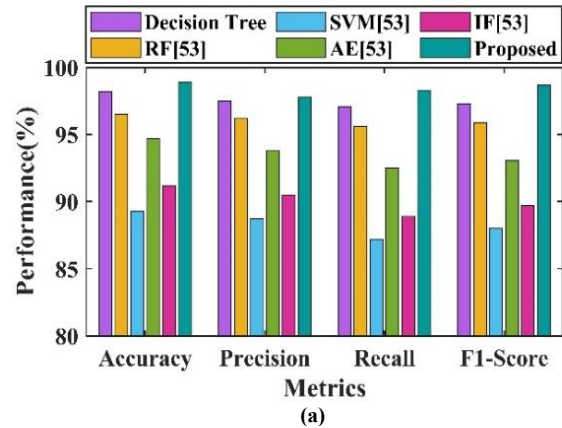


(a)

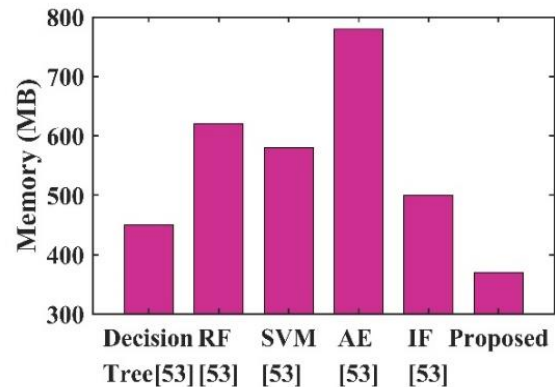


(b)

Fig. 16 Evaluation of communication stability and energy reliability based on (a) Packet drop ratio, End-to-End delay, Throughput, and (b) Energy consumption and faulty routes.



(a)



(b)

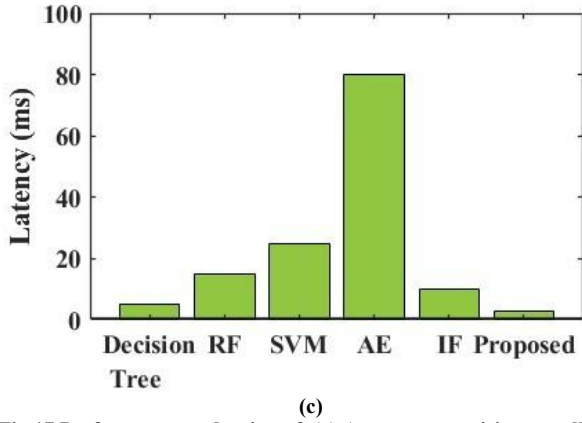


Fig.17 Performance evaluation of, (a) Accuracy, precision, recall and F1-Score, (b) Memory, and (c) Latency.

Figure 17 assesses the categorization and computational effectiveness of the suggested fault detection mechanism. Figure 17(a) shows that the accuracy, the precision, the recall and the F1-score are high, which is reliable in fault detection. Figure 17(b) demonstrates a lower usage of memory, which proves the lightweight nature of the model. Figure 17(c) has a smaller latency, which guarantees real-time applicability and quicker fault response of IoT-based WSNs.

Table 1. Fault detection reliability in distributed detection systems

Node's failure ratio ρ	Average number of neighbour nodes k			
	5	10	15	20
0.1	0.997	1	1	1
0.15	0.986	0.993	1	1
0.2	0.963	0.977	0.994	1
0.25	0.936	0.956	0.982	0.993
0.3	0.874	0.918	0.969	0.977

Table 1 describes the connection between a failure ratio ρ nodes and also the experimentally determined average connectedness of neighbour nodes k of a network. When, say, the failure ratio of the other node has to go from 0.1 to 0.3, the link of the network connection go down. It results in the connection dropping from $k=5$ to the value below $\rho=0.1$ to 0.874 at $\rho=0.3$. As such, for $k=10$, the values declined from 1 to 0.918, for $k=15$ from 1 to 0.969, and for $k=20$ from 1 to 0.977. A node with more neighbours probably still maintains connectivity under increasing failure rates.

Table 2. Performance evaluation of an enhanced distributed fault detection approach

Node's failure ratio ρ	Average number of neighbour nodes k			
	5	10	15	20
0.1	1	1	1	1
0.15	0.998	1	1	1
0.2	0.993	0.998	1	1
0.25	0.986	0.993	1	1
0.3	0.965	0.987	0.997	1

Table 2 shows the enhanced distributed fault detection model's accuracy as a function of node failure ratios ρ and average number of neighbouring nodes k . A clear trend shows an increase in failure ratios from 0.1 to 0.3 with a slight drop in the accuracy of only $k=5$, whereas at higher neighbour counts, or $k=10, 15, 20$, accuracy remains intact. At $k=5$, for the case $\rho=0.1$, it gets 1 accuracy, dropping down to 0.964 at $\rho=0.3$. This implies that as more neighbour nodes are considered, fault detection accuracy is enhanced, resulting in a greater robustness of the scheme against node failures. All of these results collectively affirm the superior diagnostic capabilities and computational efficiency of the modeled framework in IoT-WSN environments.

Table 3. Ablation analysis of the proposed framework

Model Variant	Fault Detection Accuracy (%)	False Alarm Rate (%)	Energy Consumption (J)	Network Lifetime (s)
CNN-BiLSTM	94.8	4.6	2.9	1850
SRA-CNN-BiLSTM	96.7	3.2	2.7	2000
SRA-CNN-BiLSTM + Conventional Classifier	97.1	2.8	2.6	2050
SRA-CNN-BiLSTM + SLRSN	98.3	2.3	2.4	2200
Proposed SRA-CNN-BiLSTM + SLRSN + 3CPO	99.0	2.0	2.3	2350

The results from the ablation analysis highlight the role of each essential component in the suggested strategy in Table 3. The purely CNN-BiLSTM model provided a fault identification accuracy of 94.8% and a false alarm rate of 4.6%, as well as an energy consumption of 2.9 Joule and a network lifetime of 1850 seconds. When adding the SRA solution, the accuracy of the SRA-CNN-BiLSTM model increased to 96.7%, while the false alarm rate dropped to 3.2%, which shows that attention-based residual feature

extraction has proven its effectiveness. The introduction of SLRSN improved the outcome of the model even further, and it achieved an accuracy of 98.3%, a false alarm rate of 2.3%, and energy consumption of 2.4 Joules. Finally, the complete model combined with 3CPO optimization achieved the best rate with the fault detection accuracy of 99.0%, the false alarm rate of 2.0% and the energy consumption of 2.3 Joule, as well as the maximum network lifetime of 2350 seconds. These results illustrate that every part is valuable individually and

demonstrate that the combination of the SRA-CNN-BiLSTM, SLRSN, and 3CPO model has great performance.

3.5. Discussion

The attribute to the superior efficiency of the proposed two-stage fault management framework, in comparison to the state-of-the-art techniques to the successful implementation of high-level deep learning models and optimization techniques. Existing techniques such as FEDLSTM, GBC, DRL, KF and optimization-based fault diagnosis methods have been shown to perform satisfactorily, yet they fall short due to feature dependence, weak capability of adaptation to changing fault patterns, and the requirement of considerable resources for computation. The proposed SRA-CNN-BiLSTM model solves the issues of performance with the help of the ability to extract the spatial and temporal information from the different sensor data streams, which helps recognize various fault patterns correctly. In addition, by using the SLRSN-based classification module with its adaptive shrinkage learning, which removes insignificant features but saves important fault-related features of the model and increases the fault classification efficiency. The implementation of the 3CPO optimizer adds additional value to the model, increasing the efficiency of the tuning of the model and preventing its premature convergence, which raises the stability of learning and performance.

4. Conclusion

The two-stage fault management framework optimally identifies, classifies, and mitigates faults by the resilience, energy efficiency of IoT-based WSNs. A combination of SRA-CNN-BiLSTM for fault detection and SLRSN for fault

classification ensures that hardware and software are well distinguished. 3CPO fine-tunes the fault classification by optimizing the learning parameters to achieve enhanced accuracy and efficiency. In addition, energy-efficient mechanisms are deployed to either reintegrate or isolate faulty nodes to prolong the network lifespan and guarantee system robustness. The result indicates that the proposed method is within 0.997 seconds. Performance analysis using the Python tool outperforms existing methods regarding fault detection accuracy 98%, hardware failure detection accuracy 0.95, FAR 0.02, new fault detections saturate at about 0.06, higher trigger time 0.2, and there are fewer detections with a low false detection rate.

The future research works will be directed towards augmenting the framework with federated learning for decentralized fault management, improving real-time adaptability, and extending applications to various IoT scenarios ranging from underwater to space sensor networks.

Acknowledgement

None

Data Availability Statements

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

- [1] Daudi Mnyanghwal, and Shamte Kawambwa, "Deployment of an IoT based Sensor Node for Faults Detection and Classification in Electrical Secondary Distribution Networks," *Cogent Engineering*, vol. 11, no. 1, pp. 1-17, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Ahmad Saeedi, Marjan Kuchaki Rafsanjani, and Samaneh Yazdani, "Energy Efficient Clustering in IoT-based Wireless Sensor Networks using Binary Whale Optimization Algorithm and Fuzzy Inference System," *The Journal of Supercomputing*, vol. 81, no. 1, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Ehsan Heidari, "A Novel Energy-Aware Method for Clustering and Routing in IoT based on Whale Optimization Algorithm and Harris Hawks Optimization," *Computing*, vol. 106, no. 3, pp. 1013-1045, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Tabassum Kanwal et al., "Energy-Efficient Internet of Things-based Wireless Sensor Network for Autonomous Data Validation for Environmental Monitoring," *Computer Systems Science and Engineering*, vol. 49, no. 1, pp. 185-212, 2025. [[CrossRef](#)] [[Publisher Link](#)]
- [5] Walid Osamy, "LBAS: Load-Balancing Aware Clustering Scheme for IoT-based Heterogeneous Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 24, no. 9, pp. 15472-15490, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Preeti Sharma et al., "NHARSO-IWSN: A Novel Hybridized Adaptive-Network-based Fuzzy Inference System with Reptile Search Optimization Algorithm-based Routing Protocol for Internet of Things-Enabled Wireless Sensor Networks," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 6293-6302, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] E. Silambarasan et al., "Energy Efficient Heterogeneous Clustering Scheme using Improved Golden Eagle Optimization Algorithm for WSN-based IoT," *International Journal of Information Technology*, vol. 17, no. 3, pp. 1753-1760, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] G. Mahalakshmi, S. Ramalingam, and A. Manikandan, "An Energy Efficient Data Fault Prediction based Clustering and Routing Protocol using Hybrid ASSO with MERNN in Wireless Sensor Network," *Telecommunication Systems*, vol. 86, no. 1, pp. 61-82, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [9] Nagarajan B, and Santhosh Kumar SVN, "An Intelligent Sailfish Optimization based Fault Diagnosis and Classification in Wireless Sensor Networks," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Abdennabi Morchid et al., "IoT-based Smart Irrigation Management System to Enhance Agricultural Water Security using Embedded Systems, Telemetry Data, and Cloud Computing," *Results in Engineering*, vol. 23, pp. 1-16, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Šarūnas Kilius et al., "Time Delay Characterization in Wireless Sensor Networks for Distributed Measurement Applications," *Journal of Sensor and Actuator Networks*, vol. 13, no. 3, pp. 1-16, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Ashutosh Kumar Rao, Kapil Kumar Nagwanshi, and Manoj Kumar Shukla, "An Optimized Secure Cluster-based Routing Protocol for IoT-based WSN Structures in Smart Agriculture with blockchain-based Integrity Checking," *Peer-to-Peer Networking and Applications*, vol. 17, no. 5, pp. 3159-3181, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Garima Thakur et al., "A Privacy-Preserving Three-Factor Authentication System for IoT-Enabled Wireless Sensor Networks," *Journal of Systems Architecture*, vol. 154, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] B. Nagarajan, and Santhosh Kumar SVN, "An Intelligent Fault Node Diagnosis and Classification Approach using XGBoost with Whale Optimization in Wireless Sensor Networks," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, pp. 1-20, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Gummarekula Sattibabu, Nagarajan Ganesan, and R. Senthil Kumaran, "IoT-Enabled Wireless Sensor Networks Optimization based on Federated Reinforcement Learning for Enhanced Performance," *Peer-to-Peer Networking and Applications*, vol. 18, no. 2, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Gianluca Tabella et al., "Bayesian Fault Detection and Localization Through Wireless Sensor Networks in Industrial Plants," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 13231-13246, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] S. Chakaravarthi et al., "IoT-based Trusted Wireless Communication Framework by Machine Learning Approach," *Measurement: Sensors*, vol. 34, pp. 1-8, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Yuanlong Li et al., "Deep-Reinforcement-Learning-based Wireless IoT Device Identification using Channel State Information," *Symmetry*, vol. 15, no. 7, pp. 1-21, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] I. Surenter, K.P. Sridhar, and Michelraj Kingston Roberts, "Enhancing Data Transmission Efficiency in Wireless Sensor Networks through Machine Learning-Enabled Energy Optimization: A Grouping Model Approach," *Ain Shams Engineering Journal*, vol. 15, no. 4, pp. 1-14, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] S. Ramalingam et al., "Performance Enhancement of Efficient Clustering and Routing Protocol for Wireless Sensor Networks using Improved Elephant Herd Optimization Algorithm," *Wireless Networks*, vol. 30, no. 3, pp. 1773-1789, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Rehan Khan, Umer Saeed, and Insoo Koo, "FedLSTM: A Federated Learning Framework for Sensor Fault Detection in Wireless Sensor Networks," *Electronics*, vol. 13, no. 24, pp. 1-20, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Ramineni Padmasree, and Aravalli Sainath Chaithanya, "Fault Detection in Single-Hop and Multi-Hop Wireless Sensor Networks using a Deep Learning Algorithm," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 13, no. 3, pp. 453-461, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Daniel Godfrey et al., "An Energy-Efficient Routing Protocol with Reinforcement Learning in Software-Defined Wireless Sensor Networks," *Sensors*, vol. 23, no. 20, pp. 1-22, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Divya Gupta et al., "EEDC: An Energy Efficient Data Communication Scheme based on New Routing Approach in Wireless Sensor Networks for Future IoT Applications," *Sensors*, vol. 23, no. 21, pp. 1-14, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Arash Heidari et al., "Assessment of Reliability and Availability of Wireless Sensor Networks in Industrial Applications by Considering Permanent Faults," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 27, pp. 1-21, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [26] Liyazhou Hu et al., "Security Enhancement for Deep Reinforcement Learning-based Strategy in Energy-Efficient Wireless Sensor Networks," *Sensors*, vol. 24, no. 6, pp. 1-14, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Aysha Munir Sheikh, and Sunil Joshi, "Improved Smart Energy-based Routing Approach for IoT Networks in Wireless Sensor Nodes," *Journal of Engineering and Applied Science*, vol. 71, no. 1, pp. 1-26, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Prasanth Aruchamy, Lavina Balraj, and K.K. Devi Sowndarya, "An Energy-Aware Link Fault Detection and Recovery Scheme for QoS Enhancement in Internet of Things-Enabled Wireless Sensor Network," *Computers and Electrical Engineering*, vol. 123, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Nassima Jihani, Mohammed Nabil Kabbaj, and Mohammed Benbrahim, "Kalman Filter based Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation," *Results in Engineering*, vol. 20, pp. 1-6, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Roopali Dogra, Shalli Rani, and Gabriele Gianini, "REERP: A Region-based Energy-Efficient Routing Protocol for IoT Wireless Sensor Networks," *Energies*, vol. 16, no. 17, pp. 1-16, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [31] Gagandeep Kaur, and Prasenjit Chanak, “An Intelligent Fault Tolerant Data Routing Scheme for Wireless Sensor Network-Assisted Industrial Internet of Things,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5543-5553, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [32] Zhiyi Xin et al., “SiamRAAN: Siamese Residual Attentional Aggregation Network for Visual Object Tracking,” *Neural Processing Letters*, vol. 56, no. 2, pp. 1-22, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [33] Wei Dai et al., “Network Intrusion Detection Method based on CNN-BILSTM-Attention Model,” *IEEE Access*, vol. 12, pp. 53099-53111, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] Xingwang Lv et al., “Self-Learning Guided Residual Shrinkage Network for Intelligent Fault Diagnosis of Planetary Gearbox,” *Engineering Applications of Artificial Intelligence*, vol. 139, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [35] Haijun Liu et al., “Multi-Strategy Enhanced Crested Porcupine Optimizer: CAPCPO,” *Mathematics*, vol. 12, no. 19, pp. 1-41, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]