An Intelligent Fault Tolerant Data Routing Scheme for Wireless Sensor Network-Assisted Industrial Internet of Things

Gagandeep Kaur, *Graduate Student Member, IEEE*, and Prasenjit Chanak, *Member, IEEE*

***Abstract*—Safety is a major concern for Industrial 4.0 where different physical parameters are monitored for avoiding uncertain events in the industry. In industries, natural calamities like fire and leakage of harmful gases can cause huge damage to life and property. An Industrial Internet of Things (IIoT) is used to monitor such natural calamities and take timely prompt actions. However, sensors in the IIoT are vulnerable to failures due to energy depletion and hardware malfunctioning. It significantly reduces the reliability of the network. This article proposes an intelligent fault-tolerant scheme where different faults within the wireless sensor network-assisted IIoT such as node fault and link fault are detected and tolerated in a timely manner. It significantly improves the reliability of the network. Extensive simulations show the out-performance of the proposed scheme in terms of average packet delivery, energy consumption, throughput, network lifetime, communication delay, and recovery speed.**

***Index Terms*—Fault tolerance, hybrid reinforcement learning (RL), industrial internet of things (IIoT), industrial wireless sensor networks (IWSN), Industry 4.0.**

I. Introduction

NOWADAYS, Industrial 4.0 has become a hot topic of research as it has revolutionized the industrial sector. In Industrial 4.0, different cutting edge technologies such as Industrial Internet of Things (IIoT), industrial wireless sensor networks (IWSNs), and machine learning are used for the advancement and automation of traditional industrial processes [1]. Industry 4.0 introduces smart machines such as autonomous robots, expert systems, and intelligent assistants. Smart machines significantly increase the productivity and safety of an industry. As per the current report [2], the global smart machine market will reach \$32.5 billion by 2027. In this industrial revolution, heterogeneous sensors are deployed to monitor different physical parameters of the industrial environment. Different sensor nodes form a wireless sensor network that can directly assist different IIoT applications. Intelligent wireless sensor networks (WSNs) find different applications in Industry 4.0 including healthcare [3], safety [4], and smart manufacturing [5]. Intelligence is introduced in IIoT with the use of artificial intelligence (AI). As per the Internet of Things (IoT) Tech news report [6], the annual growth rate of incorporation of AI in IIoT is expected to be 57.2% over the next five years.

An AI-based-intelligent WSN is used in IIoT for data gathering from the deployed heterogeneous sensor nodes [7]. Sensors are tiny with limited battery power, and they are prone to failures due to battery and hardware malfunctioning. In the network, faults lead to data loss, communication delay, less throughput, and poor network performance [8]. These faults hinder the working of IoT-based industrial applications. In the IWSN, one of the major reasons for the occurrence of faults is battery drainage which should be prevented by reducing energy consumption [9]. Clustering is an efficient technique that reduces energy consumption in IWSN. In clustering, sensor nodes are logically partitioned into different small groups called clusters. Each cluster has a cluster head (CH) that collects data from the cluster member nodes and transmits it to the sink node/base station for further processing.

There are several existing schemes for routing within the IWSNs [10], [11]. However, these schemes do not provide mechanisms for fault-tolerant routing in IWSNs [12]. Also, these schemes do not work properly in the dynamic environment where nodes turn faulty due to battery drainage and malfunctioning. Furthermore, existing approaches are unable to jointly handle different network faults due to node and link failures. Reinforcement learning (RL)-based routing provides a reliable routing for IWSNs [13]. Few existing schemes based on RL enhance the reliability but disregards the congestion, delay, and limited battery issue in IWSNs [14]. This article proposes an intelligent fault tolerance scheme for IWSN that addresses reliability, energy, delay, and congestion. It also jointly considers different network failures and is capable of assisting IIoT applications. The proposed scheme is also applied to develop a reliable fire detection application for the industry. The major contributions of this article are as follows.

Manuscript received 23 January 2022; revised 19 March 2022, 23 April 2022, and 25 June 2022; accepted 27 August 2022. Date of publication 6 September 2022; date of current version 22 March 2023. This work was supported in part by the Science and Engineering Research Board (SERB), and in part by the Department of Science and Technology, Government of India, under Grant SRG/2021/000090. Paper no. TII-22-0357. (*Corresponding author: Prasenjit Chanak.*)

Gagandeep Kaur is with the School of Computer Science Engineering and Technology, Bennett University, Greater Noida, UP 201310, India (e-mail: gagan873@gmail.com).

Prasenjit Chanak is with the Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, UP 221005, India (e-mail: prasenjit.chanak@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3204560.

Digital Object Identifier 10.1109/TII.2022.3204560

1) An intelligent fault tolerance scheme for IWSN is proposed to tolerate the nodes and link failures. It significantly increases the performance and reliability of the network.
2) An intelligent data routing scheme is proposed for a faster data gathering process in an industrial environment.
3) This article also applies the proposed scheme to develop a reliable and early fire detection system for Industrial 4.0 applications.
4) Extensive simulations demonstrate the effectiveness of the proposed intelligent fault-tolerant scheme in terms of average packet delivery, energy consumption, throughput, network lifetime, communication delay, and recovery speed.

The rest of the article is arranged as follows. Section II describes the literature review of the recent state-of-the-art mechanisms. Section III presents the network and energy model. The detailed proposed work is described in Section IV. Results and discussions are described in Section V. Finally, Section VI concludes this article.

II. Related Work

This literature review covers existing state-of-the-art approaches for fault-tolerant data routing in the WSNs. Tong et al. [15] proposed a Markov model-based fault-tolerant mechanism for WSNs. In this approach, backup CHs are selected to manage the fault of the working CHs. The major drawback of this approach is that it is unable to tolerate link failure. It reduces the overall fault tolerance performance of this scheme. Moridi et al. [16] proposed a cluster-based multipath routing protocol for fault tolerance in WSNs. In this approach, the CH and cluster member nodes are arranged in a tree structure where CH acts as a root node of the tree. Furthermore, a backup CHs-based fault-tolerant approach is used to tolerate CH failure. This approach uses majority voting and hypothesis testing to detect faulty sensor nodes within the networks. Detected faulty sensor node paths are avoided by the selection of three different disjoint paths. This approach selects disjoint paths based on energy consumption and hop count. However, this approach suffers from huge data delivery delays. Rui et al. [17] proposed a self-adaptive and fault-tolerant (SAFT) routing algorithm for WSNs. This approach uses advanced particle swarm optimization based path selection algorithm for fault tolerance. This approach considers two major parameters such as energy and distance for optimal path selection. Furthermore, this approach tolerates sensor nodes and CHs fault during the data transmission process by the selection of an optimal number of duplicate paths. The major drawback of this algorithm is that it suffers from huge message overhead and poor network throughput. Muhammed et al. [18] proposed a fault-tolerant routing algorithm for IoT networks. In this algorithm, multiple paths are selected and prioritized based on energy and distance. It uses dynamic source routing and vice CHs for optimal data routing and fault tolerance. However, this approach suffers from the huge message overhead and poor network lifetime. Hedar et al. [19] proposed a virtual backbones-based fault tolerance protocol for WSNs. In this protocol, virtual backbones are constructed based on connectivity, coverage, and cardinality. Furthermore, scheduling between multiple backbones is performed periodically to avoid the failure of sensor nodes. This approach uses a huge number of control messages to manage multiple backbones of the network. It significantly reduces network lifetime and increases data delivery delays. Li et al. [20] proposed a best and worst ant system (BWAS) based multipath fault tolerance algorithm for WSNs. This approach selects multiple paths for data transmission based on energy consumption and data transmission delay. Furthermore, BWAS-immune mechanism-based multipath reliable transmission algorithm (BIM2RT) is used to increase data transmission reliability. The main drawback of this algorithm is that it suffers from poor network throughput. Menaria et al. [21] proposed a Node-Link Failure Fault Tolerance (NLFFT) mechanism for WSNs. In this approach, node failure is tolerated considering three major parameters such as battery power, distance, and fault index. Furthermore, an improved quadratic minimum spanning tree is also constructed for optimal data transmission within the networks. The main drawback of this approach is that it consumes huge energy to manage and maintain a quadratic minimum spanning tree. In a highly faulty environment, it reduces the network lifetime and overall performance of the network.

The above existing state-of-the-art approaches focus to tolerate only one type of fault among node and link failures [15], [17]. However, Industrial 4.0 applications require an efficient fault tolerance scheme that can jointly detect all faults. Furthermore, Industrial 4.0 applications also demand a high-performance network in terms of reliability, low energy consumption, data delivery delays, and high fault recovery speed [21], [22]. The existing approaches are unable to fulfill all these requirements for Industrial 4.0 applications. This article proposes an intelligent fault-tolerant data routing scheme for high-performance IoT-based WSN. The proposed intelligent fault-tolerant data routing scheme fulfills the aforesaid critical requirement for real-time Industrial 4.0 applications. The proposed scheme is capable to tolerate node fault, as well as link fault that enhances the reliability, throughput, and lifetime of the networks. Furthermore, the proposed intelligent fault tolerance mechanism uses hybrid RL that effectively handles the dynamic environment due to sensor node failures. The hybrid RL whale optimization algorithm (RLWOA) is executed within the sink node to manage node and link failure. This reduces the burden of computational overhead on the resource-constrained sensor nodes [7], [23], [24].

III. Network and Energy Model

The $\mathcal{N}$ number of sensor nodes $s_1, s_2, ..., s_{\mathcal{N}}$ are randomly deployed in $\mathcal{A} \times \mathcal{A}[m^2]$ monitoring area. These sensor nodes have the same initial energy and they are static. The received signal strengths are used to measure the distance ($\mathcal{D}_{ij}$) between sensor node pairs. If $\mathcal{D}_{ij}$ is less than the transmission range then sensor node pairs can communicate. The sink has high computational power. It has unlimited energy and a large memory capacity. The radio model used in [25] is adopted to evaluate the energy

TABLE I
NOTATIONS AND DESCRIPTIONS

Notation	Description
γ_o	Threshold distance
ϵ_{efs}	Free space energy model
ϵ_{amp}	Energy consumed for multipath energy model
$\mathcal{E}_{\text{elc}}$	Energy required in switching on and off the transmitter and receiver circuitry
$\vec{A}$, $\vec{C}$	Coefficient vectors
D/	Distance between prey and its best solution
b	Constant for shape logarithmic spiral
l	Random value ∈ [-1, 1]
X_{rand}	Random position vector
s_t	Current state
s_{t+1}	Next state
a_t	Current action
λ	Learning rate value
Γ	Discount factor
r_{t+1}	Immediate reward
Q_{t+1}	Q-value for the next state
σ	Weighting parameter
ϕ_{s_i}	Residual energy of i^{th} sensor node
Ω_{s_i}	Initial energy of i^{th} sensor node
$\delta_{(i_n)}$	Euclidean distance between i^{th} sensor node and neighbor node
$\delta_{(i_f)}$	Euclidean distance between i^{th} sensor node and farthest neighbor node
$\mathcal{N}$	Number of sensor nodes
$\Re_m$	Residual energy of candidate forwarder node
ϑ_m	Traffic rate of candidate forwarder node
ρ_m	Transmission power of candidate forwarder node
$\mathcal{Q}_m$ and $\mathcal{T}_m$	Queuing time and transmission time of candidate forwarder node which are updated after each round.
φ_m	Service rate of candidate forwarder node

requirement of the sensor nodes. The $\mathcal{E}_{tx}$ is the energy required to transmit β b of data which is calculated by the following equation:

$$\mathcal{E}_{tx} = \begin{cases} \left(\mathcal{E}_{\text{elc}} + \epsilon_{efs} \times \gamma^2\right) \beta_i & \text{if } \gamma < \gamma_0 \\ \left(\mathcal{E} + \epsilon_{amp} \times \gamma^4\right) \beta_i & \text{if } \gamma \geq \gamma_0 \end{cases} \tag{1}$$

where γ is the distance between source and destination nodes. Let $\mathcal{E}_{rx}$ energy is required to receive β b of data which is calculated as follows:

$$\mathcal{E}_{rx} = \beta \times \mathcal{E}_{\text{elc}} \tag{2}$$

$$\gamma_o = \sqrt{\epsilon_{efs}/\epsilon_{amp}}. \tag{3}$$

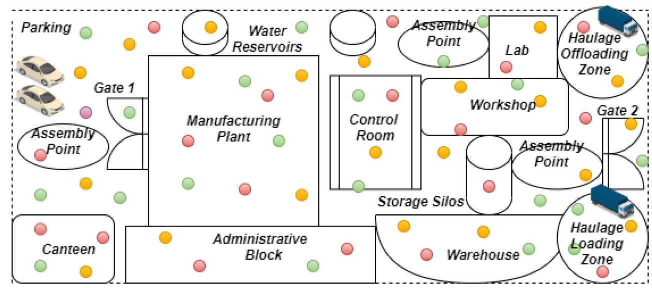


Fig. 1. Smart industry network.

Table I summarizes the list of notations with their descriptions.

IV. PROPOSED WORK

Fig. 1 shows a typical industrial environment which includes a manufacturing plant, workshop, assembly point, haulage loading, offloading zones, canteen (kitchen), etc. In the smart industry, multisensors including gas, smoke, and temperature are deployed randomly for fire detection. Multiple sensors help to enhance the accuracy and reliability of the fire detection system. Table II shows the different threshold limits for different zones. If the data generated by sensors cross the threshold levels then unwanted events such as fire or harmful gas leakage are detected. The threshold values for the manufacturing plant and kitchen are comparatively higher than the other zones. Higher values of threshold limits are set due to the higher intensity of smoke and temperature in that specific zones. The fault occurrence probability of a particular zone is dependent on the intensity of smoke, gas, and temperature in that zone which is shown in Table II. The proposed intelligent fault tolerant fire detection scheme is divided into two phases namely the intelligent fault-tolerant mechanism for node failure and the intelligent fault-tolerant mechanism for link failure. The experimental results which are shown in the result section demonstrate that the proposed intelligent fault-tolerant scheme provides a highly reliable data routing in IoT-based Industrial 4.0 applications.

A. Intelligent Fault-Tolerant Mechanism for Node Failure

In this section, the clustering and fault tolerance mechanism for IWSNs is discussed. After different gas, smoke, and temperature sensor nodes deployment, the sink broadcasts a *WAKE-UP* message in its communication range. The *WAKE-UP* message contains the sink ID and location information of the sink. If any sensor node receives *WAKE-UP* message then the sensor node responds with the *RESPONSE* message. The *RESPONSE* message contains sensor node ID and location information. If the sensors are not within the range of the sink and do not receive a *WAKE-UP* message then it broadcasts an *AID* message. The *AID* message includes the sensor node ID and location information. The sensor node that receives the *AID* message will reply with the *ANSWER* message. The *ANSWER* message contains sensor node ID, location information, and

TABLE II
THRESHOLD FOR SENSORS IN SMART INDUSTRY

Area/ Zone	*Smoke sensor (ppm)*	*Gas sensor (ppm)*	*Temperature sensor* (°C)	*Probability of fault*
Parking, haulage loading and offloading zone	160	160	45	Medium
Assembly point, administrative block	140	140	45	Low
Lab, control room	140	140	41	Low
Warehouse, water reservoirs, storage silos	140	140	43	Low
Canteen (kitchen), manufacturing plant	200	200	50	High

residual energy. After receiving the *ANSWER* message, the sensor node joins the intermediate sensor node with greater residual energy. Thus, the sink collects all the sensor node's IDs and location information through intermediate sensors. Furthermore, *K*-means clustering is performed by the sink that groups the sensors into *K* number of clusters. Cluster formation is performed depending on the minimum distance between sensor nodes. Let $l_1, l_2, l_3, \ldots .l_{\mathcal{N}-1}, l_{\mathcal{N}} \in \mathcal{L}$ are the location of the $\mathcal{N}$ sensor nodes and $\zeta_1, \zeta_2, \zeta_3, \ldots .\zeta_{K-1}, \zeta_K \in \mathcal{C}$ are the target cluster head locations. Each sensor node location is computed based on x and y coordinates where (x_r, y_s) represent sensor node location l_i. Target CHs locations are also computed based on x and y coordinates where (x_p, y_q) is represented by ζ_j. Each sensor $l_i \in \mathcal{L}$ is associated with the nearest centroid CH in set $\mathcal{C}$ by applying the shortest distance which is defined as follows:

$$\mathcal{L}^{\{\mathcal{C}_j\}} = \arg\min \|l_i - \zeta_j\|^2 \tag{4}$$

where $i = 1, \ldots, N$ and $j = 1, \ldots, K$.

The cluster mean is calculated for each cluster $\mathcal{L}^{\{\mathcal{C}_j\}}$ using the following equation:

$$\mathcal{M}_j = \frac{\sum_{i=1}^{v} l_i}{v}, j = 1, \ldots, K. \tag{5}$$

Then, with the help of mean $\mathcal{M}$ from above equation optimal locations of cluster heads $\mathcal{C}'$ are calculated as follows:

$$\{\mathcal{C} \rightarrow \mathcal{C}'\} = \{\zeta_1 = \mathcal{M}_1, \zeta_2 = \mathcal{M}_2, \ldots, \zeta_K = \mathcal{M}_K\}. \tag{6}$$

$\Delta\mathcal{C}$ is the difference between optimal and previous cluster heads locations. It is calculated as follows:

$$\Delta\mathcal{C} = \mathcal{C}' - \mathcal{C} = \sum_{i=1}^{K} \mathcal{M}_i - \zeta_i. \tag{7}$$

The equation (5), equation (6), and equation (7) are iteratively computed until $\Delta\mathcal{C} = 0$, which is the convergence criteria. Finally, when the convergence criterion is achieved then optimal clusters are formed.

In the proposed scheme, the *HELLO* message is transmitted periodically to check the fitness of the CHs. The *HELLO* message is only transmitted to CHs by the sink node which significantly reduces the message overhead within the networks. The *HELLO* message contains the sink ID, location information, and fitness status of the sink. This work assumes that the sink has an unlimited energy supply and computational capacity. Therefore, the sink always set its fitness status as "OK." If any CH receives the *HELLO* message then it replies with a *REPLY* message. The *REPLY* message includes CH ID, location information and residual energy, and its fitness status. Therefore, the *HELLO* message is not flooded within the networks and it is optimally used to check the fitness of other nodes. The number of *HELLO* messages transmitted by the sink node is dependent on the environmental condition. In highly faulty environments such as industrial environments, sink nodes are frequently required to check the fitness status of the CHs compared to a less faulty environment such as a home monitoring system. The cost of periodic checking of the fitness status of the CHs can be optimized. The objective of this optimization problem is to minimize the cost of message overhead. It is formulated as follows:

$$\mathrm{C}_{\mathrm{mo}} = K * \frac{N_l}{t} * P_t \tag{8}$$

Algorithm 1: Intelligent Fault-Tolerant Mechanism for Node Failure.

Input: Sensor nodes: $s_1, s_2 \ldots s_{N-1}, s_N$
Clusters: $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3 \ldots \mathcal{C}_k$
Cluster Heads: $\mathcal{CH}_1, \mathcal{CH}_2, \mathcal{CH}_3 \ldots \mathcal{CH}_k$
Sink

Sink broadcasts a *WAKE-UP* message
if Sensors receive the *WAKE-UP* message **then**
  Sensors reply with *RESPONSE* message
end if
The sink forms the optimal clusters using K-means
Sink broadcasts a *HELLO* message periodically
HELLO message contains sink ID, location, and fitness status
if Sensors receives the *HELLO* message **then**
  Sensors responds with *REPLY* message
end if
REPLY message includes sensor ID, location, residual energy
if *REPLY* is not received by sink **then**
  Sink marks CH node as Hardware faulty
else if *REPLY* is received by sink and energy drops to a threshold value **then**
  Sink marks CH node as Battery faulty
  Replace faulty *CHs* using $\mathcal{RLWOA}$ (Algorithm 2)
  Apply fitness function using (11), 9, 10
  Obtain new *CHs*
end if

subject to

$$Constraint : t < t_{\text{th}}$$

where C_{mo} is the maximum cost of message overhead and K is number of CHs, N_l is network lifetime. t is periodic time slot, P_t is probability of fault occurrence in a zone, and t_{th} is the maximum threshold limit of the time duration to check the fitness of other nodes. If the cluster head is not within the range of the sink and does not receive a *HELLO* message, then it broadcasts *HELP* message. The *HELP* message contains the CH ID, location information, residual energy, and fitness status. The CH that receives the *HELP* message will reply with the acknowledgment *ACK* message. The *ACK* message contains CH ID, location information, and residual energy. After receiving the *ACK* message, far away CH communicates to the sink node with help of intermediate CH.

After some time, if the sink node does not receive a reply from any CH then the sink node marks it as a hardware faulty node. In this work, the hardware faulty node is defined as the transmitter, receiver, or microcontroller unit failure of the sensor module. Also, the sink may not receive data from the CHs due to link failure. The link failure is defined as the failure of data transmission from the CHs to other intermediate CHs. The proposed scheme applies an intelligent fault-tolerant mechanism for link failures which is described in Section IV-B. Furthermore, when the sink receives the fitness status of CHs then it checks its residual energy. If the residual energy of CH drops to a threshold value, the sink node marks the CH as a battery faulty. The sink starts the recovery process when any of the faults are detected in the CHs. The new optimal CHs are selected by RLWOA-based fault-tolerant CHs selection mechanism (Algorithm 1). The new optimal CHs are selected based on residual energy and node degree that significantly increase the network lifetime. RLWOA-based new optimal CHs selection mechanism depends on the fitness parameters. The fitness parameters considered for optimal CHs selection are described as follows.

The first fitness parameter selects the maximum residual energy node as a CH. If CH has a high energy level, then the network remains stable for a longer duration. This increases the network lifetime. The first fitness parameter is defined as follows:

$$\wp_1 = \frac{1}{\sum_{i=1}^{N} \frac{\Omega_{s_i}}{\phi_{s_i}}}. \tag{9}$$

The second fitness parameter selects the maximum degree node as a CH from the candidate CH nodes. The second fitness parameter is described as follows:

$$\wp_2 = \left(\frac{\sum_{i=1}^{\mathrm{N}} \delta_{(\mathrm{i_n})}}{\mathrm{N}} \times \frac{1}{\delta_{(\mathrm{i_f})}} \right). \tag{10}$$

The problem of new optimal CHs selection for fault management is formulated as

$$minimize \mathcal{F}_{CH_{\text{new}}} = \wp_1 \times w_1 + \wp_2 \times (1 - w_1). \tag{11}$$

Algorithm 2 shows the pseudocode of the RLWOA-based fitness function optimization mechanism. Mirjalili [26] proposed a whale optimization algorithm that is inspired by the hunting mechanism of whales. It uses a spiral bubble-net attacking mechanism to find the prey. In the proposed fault-tolerant CHs selection mechanism, the preys are the new CHs that replace the faulty CHs. The whales are the agents which search for prey. The agents are the sensor nodes that search for optimal CHs. Each whale acts as an agent whose position is computed by the following equations:

$$\vec{D} = \left| \vec{\mathrm{C}} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \tag{12}$$

Algorithm 2: Reinforcement learning-based Whale optimization algorithm (RLWOA).

Input: Intial values of: a, A, C, Q-value
  Max number iterations :$Iter_{max}$
while $(Iter < Iter_{max})$ **do**
  Evaluate the fitness of each search agent (16)
Switch A
Case 1 : **A** $>=$ **1**
  if $(Expr_V > Expt_V)$
    Calculate new fitness value (16)
    if $(Fitness_{new} < Fitness_{curr})$
    Reward = 1
    else
    Reward = - 1
    end if
  else
    Calculate new fitness value using (21)
    if $(Fitness_{new} < Fitness_{curr})$
    Reward = 1
    else
    Reward = -1
    end if
  end if
  Update Q-value using (19)
  break
Case 2 : **A** $<$ **1**
  if $(Expr_V > Expt_V)$
    Calculate new fitness value (16)
    if $(Fitness_{new} < Fitness_{curr})$
    Reward = 1
    else
    Reward = -1
    end if
  else
    Calculate new fitness value using (21)
    if $(Fitness_{new} < Fitness_{curr})$
    Reward = 1
    else
    Reward = -1
    end if
  end if
  Update Q-value using (19)
  break
end while

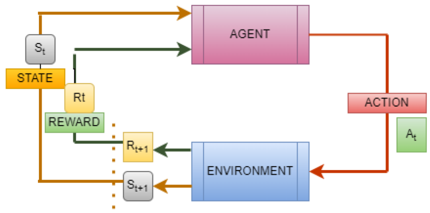


Fig. 2. Reinforcement learning.

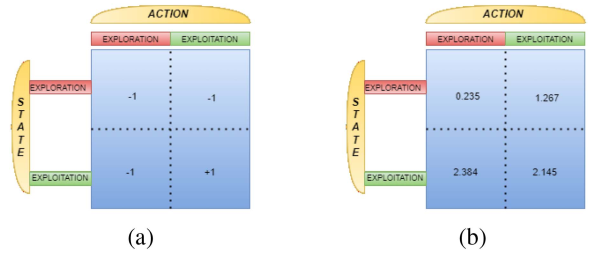


Fig. 3. Sample of reward table and Q table. (a) Reward Table. (b) Q Table.

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \tag{13}$$

where $\vec{X}^*(t)$ represents the position vector of prey (new CH) and $\vec{X}(t)$ is the position vector of the sensor node. The $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{14}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{15}$$

where components of $\vec{a}$ are linearly decreased from 2 to 0 over the iterations. r_1, r_2 are random vectors $\in$ [0, 1]. The value of coefficient $\vec{c}$ varies between 0 to 2, which represents the random behavior of prey. In (11), the fitness function is evaluated based on the fitness parameters of each search agent. The fitness parameters $\wp_1$ and $\wp_2$ are described in (9) and (10). The shrinking encircling and spiral updating positions are calculated by the following equation:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t+1) - \vec{A} \cdot \vec{A} & \text{if } p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \tag{16}$$

$$\overrightarrow{\mathrm{D}} = \left| \overrightarrow{\mathrm{C}} \cdot \overrightarrow{\mathrm{X}}_{\mathrm{rand}} - \overrightarrow{\mathrm{X}} \right| \tag{17}$$

$$\overrightarrow{\mathrm{X}}(\mathrm{t}+1) = \overrightarrow{\mathrm{X}}_{\mathrm{rand}} - \overrightarrow{\mathrm{A}} \cdot \overrightarrow{\mathrm{D}}. \tag{18}$$

The RL algorithm is used for learning optimal actions by agents in a complex environment [27]. Fig. 2 shows the layout of the RL procedure. The sink is trained and performs experienced-based actions using hybrid RL. It chooses optimal CHs. In a dynamic environment, the network topology is continuously changed due to CHs energy exhaustion and failures. Therefore, a hybrid algorithm is used that combines RL and metaheuristic algorithms. The hybridization integrates the advantage of both algorithms and mitigates their drawbacks. In metaheuristic algorithms, prior deep knowledge is used to balance exploration and exploitation. Exploration indicates the searching for optimal solutions over the whole sample space. Exploitation indicates searching for the most optimal solution from optimal solutions which are found during the exploration. Hybrid metaheuristic algorithms are more dynamic in nature as compared to meta-heuristics algorithms due to value-based RL. RL algorithms are more successful to find new local areas in a balanced way. However, the RL algorithm takes more time in the learning process. Therefore, the hybridization of metaheuristic and RL algorithms shows better performance in balancing and switching between exploration and exploitation. Furthermore, hybridization also shows fast convergence behavior as compared to conventional greedy and softmax-based Q-learning methods [28], [29].

In the proposed scheme, an agent can be in explore state or exploit state. An agent can perform the following four actions using these two states.

1) The agent is in explore state and continues to be in the explore state.
2) The agent is in explore state and switches to an exploit state.
3) The agent is in an exploit state and continues to be in an exploit state.
4) The agent is in an exploit state and switches to the explore state.

In each iteration, an agent changes its state by taking the best action which is decided by the highest Q table value. After execution of an action by an agent, it is assigned a reward [30]. The agent updates its Q-values for the next iteration using the Bellman equation which is described as follows:

$$\begin{aligned} Q_{\langle t+1)}\left(s_t, a_t\right) &\leftarrow Q_t\left(s_t, a_t\right) \\ &+ \lambda\left[r_{t+1} + \Gamma \operatorname{Max} Q_{t(s_{t+1},a)} - Q_t\left(s_t, a_t\right)\right]. \end{aligned} \tag{19}$$

The Q-table is a matrix that is initially set to zero which signifies that the agent has no experience at an initial stage. Agent updates the Q-value using the Bellman equation that suggests an agent has gained experience over iterations. The reward table is a matrix that contains +1 or −1 values. It describes reward or penalty factors. Fig. 3 shows a sample of the reward table and Q table. In the proposed RLWOA, |A| along with Q-value is used to decide whether to switch between explore state or exploit state. In Algorithm 2, lines 4 and 22 indicate the dependency of the |A| value which is obtained from the metaheuristic algorithm WOA. The inner loop lines 5–21 and 23–39 are used to compare exploration and exploitation values obtained from Q-learning. The Q-value in the exploration state is defined as the exploration value. The Q-value in the exploitation state is also termed as exploitation value [20]. If the exploration value is less than the exploitation value which is evaluated using Q-value then new fitness value is calculated as follows:

$$\vec{X}(t+1) = \begin{cases} \sigma \cdot \vec{X}*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \sigma \cdot \left(D' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}*(t)\right) & \text{if } p \geq 0.5. \end{cases} \tag{20}$$

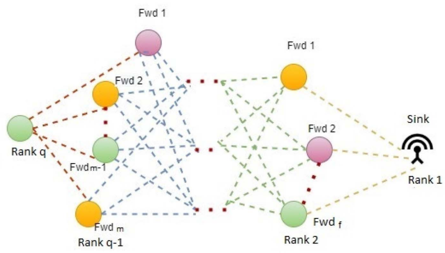


Fig. 4. Fault tolerant forwarder node selection mechanism.

B. Intelligent Fault Tolerant Mechanism for Link Failure

This section presents an intelligent fault-tolerant mechanism for link failure. For data routing, the sink assigns itself a Rank (sink) $=1$ and broadcasts an announcement *ANN* message to CHs in its communication range. The ANN message contains sink ID, location information, and rank. If CH_i receives *ANN* message then it updates rank one greater than the sink, i.e., *Rank*(CH_i)*= Rank(sink) + 1*. Furthermore, CH_i defines the sink node as a parent forwarding node *(PFN)*, i.e., *PFN*(CH_i) *= sink*. Similarly, all CHs within the communication range of the sink increases their rank one more than the sink node and set the sink as their parent forwarding node (Fig. 4). Iteratively, the CH_i broadcasts a collection *COLL* message to the other CHs in range $R_{\max}$. The $R_{\max}$ is the communication range of the sensor node. The $R_{\max}$ calculation depends on transmitting power. In the proposed work, the maximum value of $R_{\max}$ is 50 m which depends on the hardware configuration of the node. The *COLL* message contains CH ID, rank, location information, and residual energy. If CH_j receives the *COLL* message from CH_i then it updates its rank one greater than the CH_i and assigns it as the forwarding node, i.e., *Rank*(CH_j)*=Rank*(CH_i) *+ 1* and *PFN*(CH_j) $=CH_i$. If CH_j does not receive the *COLL* message due to link failure then it searches optimal alternative path. Link failure occurs due to hardware faults, moving obstacles, etc. If cluster heads links are broken due to link failures, then the optimal alternate path is searched by applying RLWOA in centralized mode [23]. The sink is trained based on the experienced-based RL and applies the reward-based strategy to find the optimal forwarding node. RL-WOA considers residual energy, the number of *COLL* messages received from the CHs, and communication delay to select the best parent node from the *PFN* set. Therefore, RLWOA-based *PFN* node selection mechanism effectively manages link failure during the data routing process. This process is repeated recursively until all the CHs are assigned *PFN*. The fitness parameters considered for optimal *PFN* selection are described as follows.

The first fitness parameter is used to select a *PFN* among multiple forwarder nodes that maximize the network lifetime. It is formulated as follows:

$$\varrho_1 = \left\| \frac{1}{\Re_m/(\vartheta_m \times \rho_m)} \right\|. \tag{21}$$

Algorithm 3: Intelligent Fault Tolerant Mechanism for Link Failure.

Input: Cluster Heads: $\mathcal{CH}_1, \mathcal{CH}_2, \ldots..\mathcal{CH}_{K-1}, \mathcal{CH}_K$ Sink

Rank(Sink)=1
Sink broadcasts *ANN* message
ANN message contains sink ID, location, Rank
Rank(CH_i)*= Rank(sink) + 1*
PFN(CH_i) *= sink*
CH_i broadcasts *COLL* message
Rank(CH_j)*= Rank*(CH_i) *+ 1*
PFN(CH_j) $= CH_i$
CH_i selects $PFN \in CH(fwd_1, fwd_2, \ldots., fwd_m)$ using RLWOA(Algorithm 2)
Apply fitness function using (24), (21), (22), (23)
Obtain *PFN*

The second fitness parameter is used to minimize the communication delay. It is calculated as follows:

$$\varrho_2 = \mathcal{Q}_m + \mathcal{T}_m. \tag{22}$$

The third fitness parameter is used to minimize the traffic congestion which is calculated as follows:

$$\varrho_3 = \frac{\vartheta_m}{\varphi_m}. \tag{23}$$

The optimal *PFN* selection depends on the fitness parameters involved to calculate the fitness value which is described as follows:

$$minimize \mathcal{F}_{\text{fwd}_{\text{new}}} = \varrho_1 \times \omega_1 + \varrho_2 \times \omega_2 + \varrho_3 \times \omega_3 \tag{24}$$

where $\omega_1 + \omega_2 + \omega_3 = 1$.

The pseudocode of RLWOA based fault-tolerant data routing scheme is shown in Algorithm 3. Fig. 5 shows the flow diagram of the proposed RLWOA. In the proposed fault-tolerant data routing mechanism, preys are *PFNs* and whales are agents. The agents are the candidate forwarder CHs. Each whale acts as an agent whose position is calculated by (12) and (13), where $\vec{X}^*(t)$ represents the position vector of the prey (*PFN*) and $\vec{X}(t)$ is the position vector of candidate forwarder nodes. $\vec{A}$ and $\vec{C}$ are calculated using (14) and (15). In (24), the fitness function is evaluated based on the fitness parameters of each search agent. The fitness parameters ϱ_1, ϱ_2, and ϱ_3 are described in (21), (23), and (23). The shrinking encircling and spiral updating positions are selected based on (16). The RL algorithm is used for learning optimal actions by agents in a dynamic environment. The sink is trained and performs experienced-based actions to choose optimal PFN. In a dynamic environment, the value-based model-free RL works most efficiently. The Q-learning takes the best action, i.e., whether to explore or to exploit. It selects the next state from the current state using (19). If the exploration value is less than the exploitation value which is evaluated using Q-value then the new fitness value is calculated by (20).

Lemma 1: The message complexity of the proposed scheme is $\approx O(\mathcal{N} + \mathcal{N} + r \times K + r \times K + K + a + b + w + z)$.

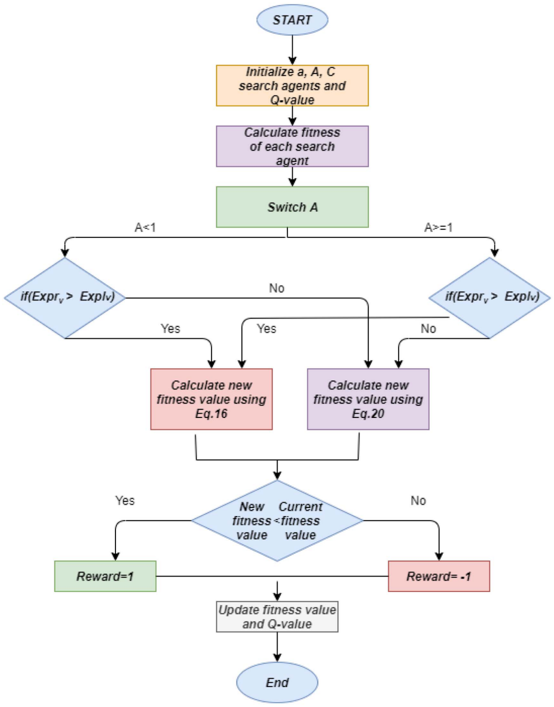


Fig. 5. Flowchart of proposed RLWOA.

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Monitoring area	200 x 200 m^2
Sensor nodes	60
Sensor nodes energy	0.5J
E_{elc}	50 nJ/bit
ϵ_{efs}	10 pJ/bit/m^2
ϵ_{amp}	0.0013 pJ/bit/m^4
$Iter_{\max}$	200

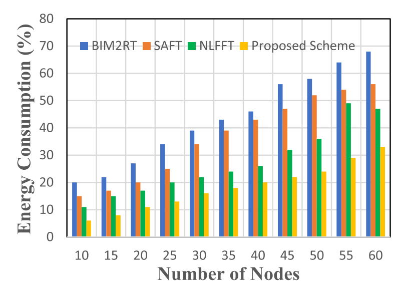


Fig. 6. Energy consumption.

Proof: In the proposed scheme, the sink takes O($\mathcal{N}$) to broadcast the *WAKE-UP* message once to the sensor nodes. The sensor nodes replies with O($\mathcal{N}$) *RESPONSE* messages. The sink broadcasts periodically *HELLO* message with O($r \times K$) complexity where *r* is the number of times the *HELLO* message is sent. The CHs reply with *REPLY* message with message complexity O($r \times K$) on receiving the *HELLO* message. The *AID* message is send by *a* number of isolated sensor nodes which receive *ANSWER* by *b* number of sensor nodes. These isolated sensor nodes message complexity is *O(a+b)*. The *HELP* message is sent by *w* number of isolated CHs that receive *ACK* by *z* number of CHs. These isolated CHs message complexity is *O(w+z)*. In data routing mechanism, the sink broadcasts *ANN* messages to the CHs and each CH broadcasts *COLL* messages to other CHs with total message complexity O(*K*). Therefore, the total message complexity of the proposed scheme is $\approx O(\mathcal{N} + \mathcal{N} + r \times K + r \times K + K + a + b + w + z)$. ■

Lemma 2: The total energy consumption by the network is $\mathcal{E}_m$+$\mathcal{E}_n$+$\mathcal{E}_o$.

Proof: In the proposed scheme, there are *N* number of sensor nodes which forms *k* number of clusters. There are an average $\mathcal{N}/k$ number of sensor nodes in each cluster. The energy requirement of each cluster member node for transmitting Φ b data to its CH is $\mathcal{E}_m = \Phi E_{\text{elec}} + \Phi \epsilon_{efs} d^2_{toCH}$. The energy required by CH for receiving and aggregating message in intracluster routing is $\mathcal{E}_n = \Phi(E_{rx}(\frac{N}{k} - 1) + E_{\text{agg}}(\frac{N}{k}))$, where E_{agg} is energy required to aggregate the data. In intercluster routing, the energy required by CH for transmitting Φ b data to next-hop CH is $\mathcal{E}_o = \xi \times E_{rx} + \xi \times E_{tx}(CH_{nh})$, where ξ is incoming data packets from other CHs and CH_{nh} is next-hop CH. Thus, total energy consumed by the network is $\mathcal{E}_m + \mathcal{E}_n + \mathcal{E}_o$. ■

V. RESULTS AND DISCUSSION

This section describes simulation results and the testbed which are used to analyze the performance of the proposed scheme.

A. Simulation Results

The simulation is performed in ns3.29 running on Ubuntu 20.04. Initially, all sensor nodes are fault free. In this network simulation experiment, dynamic changes in the environment occur due to battery drainage and malfunctioning of sensor nodes. 10% faulty sensor nodes are injected during the simulation. The simulation results of the proposed fault tolerance scheme are compared with the BIM2RT [20], SAFT [17], and NLFFT [21]. The efficiency of the proposed fault-tolerant data routing scheme is analyzed in terms of average packet delivery, energy consumption, throughput, network lifetime, communication delay, and recovery speed. Table III enlists various simulation parameters.

Fig. 6 shows the average energy consumption in different numbers of nodes. The energy consumption of the proposed scheme is reduced by 55.4% than BIM2RT, 53.3% than SAFT, and 51.5% than NLFFT. The proposed scheme outperforms in terms of energy consumption due to the RLWOA-based fault tolerance data routing scheme. RLWOA-based fault tolerance scheme for node and link failure is done by the minimum number of message exchanges between the sensor nodes. It significantly reduces the energy consumption of the proposed scheme.

Fig. 7 shows the throughput of the network in different numbers of nodes. The throughput of the proposed scheme is increased by 40.4% than BIM2RT, 39.3% than SAFT, and 38.2% than NLFFT. The proposed RLWOA-based intelligent fault tolerance data routing mechanism for link failure uses an

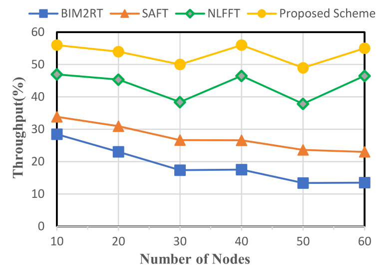


Fig. 7. Throughput.

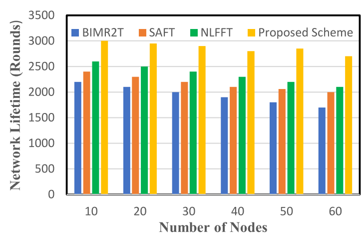


Fig. 8. Network lifetime.

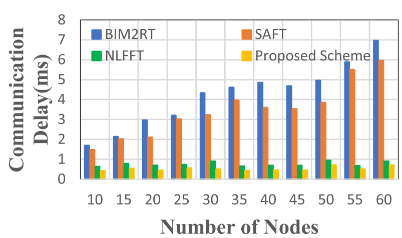


Fig. 9. Communication delay.

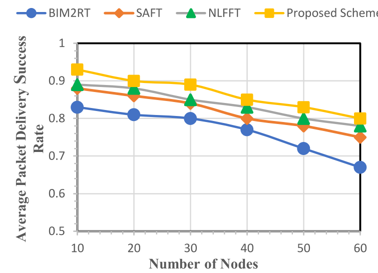


Fig. 10. Average packet delivery.

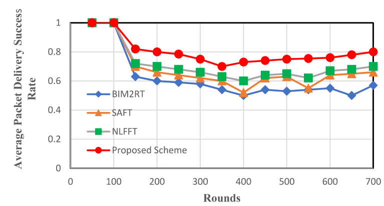


Fig. 11. Recovery speed.

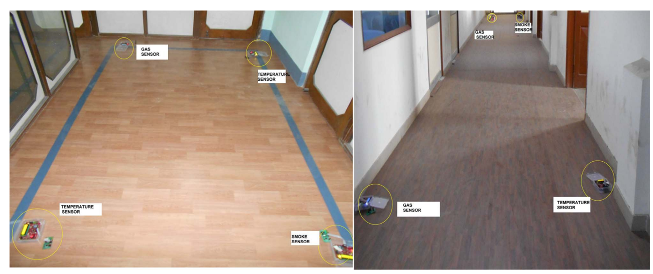


Fig. 12. Testbed experiment.

alternative path for successful data transmission from sensors to sink. Also, it effectively manages nodes and links failure. Hence, it improves the network throughput.

Fig. 8 demonstrates the network lifetime in the different number of nodes. The network lifetime of the proposed scheme is increased by 65.4% than BIM2RT, 62.3% than SAFT, and 60.4% than NLFFT. The proposed scheme reduces the energy consumption of the deployed sensor nodes with the help of minimum message exchanges between the sensor nodes. It helps to improve network lifetime.

Fig. 9 depicts the communication delay in the different number of nodes. The communication delay of the proposed scheme is reduced by 34.4% than BIM2RT, 32.3% than SAFT, and 31.5% than NLFFT. In the proposed scheme, the communication delay is reduced due to intelligent fault-tolerant data transmission where an optimal alternative path is selected for link failure.

Fig. 10 shows the average packet delivery in different numbers of nodes. The average packet delivery success rate of the proposed scheme is increased by 39.4% than BIM2RT, 38.3% than SAFT, and 36.7% than NLFFT. The proposed fault tolerance data routing mechanism for link failure efficiently selects the optimal data routing path with the help of a hybrid RL-based mechanism. It significantly enhances average packet delivery.

Fig. 11 illustrates the recovery speed in the different number of rounds. The recovery speed of the proposed scheme is increased by 37.6% than BIM2RT, 36.3% than SAFT, and 35.8% than NLFFT. In the proposed scheme, a hybrid RL-based fault tolerance mechanism helps in speedy recovery.

B. Testbed

Fig. 12 shows the real testbed where the proposed scheme has been implemented and tested. In this real testbed, eight different sensor nodes are deployed in a $30 \times 20[m^2]$ area within a building. The eight different sensor nodes contain three gas sensors, three temperature sensors, and two smoke sensors. The initial values of temperature sensors are 26 °C, gas sensors are 16 ppm, and smoke sensors are 60 ppm. The sink communicates with sensors using the ZigBee protocol. The fault-tolerant data routing-based fire detection system is tested by igniting the burning material to release gas and smoke in the testbed environment. Also, two sensors were turned faulty to measure the efficacy of the proposed system. The readings of temperature, smoke, and gas sensors change as the fire spreads. The alarm generates when the sensor nodes reading crosses the threshold limit.

Fig. 13(a) shows the average energy consumption in different numbers of nodes. The energy consumption of the proposed

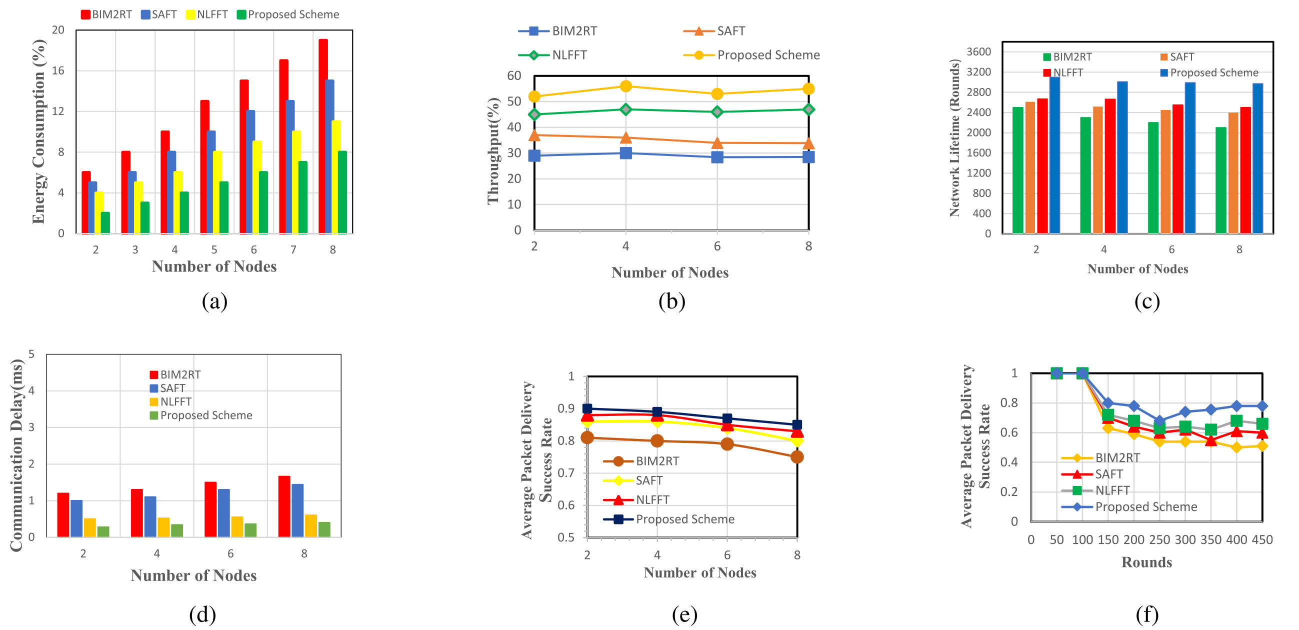


Fig. 13. Testbed results. (a) Energy consumption. (b) Throughput. (c) Network lifetime. (d) Communication delay. (e) Average packet delivery. (f) Recovery speed.

scheme is reduced by 54% than BIM2RT, 52% than SAFT, and 50% than NLFFT. As described in the simulation result, the proposed scheme also shows better energy efficiency in the real experiment due to an optimal number of CHs selection and minimum message overhead. Fig. 13(b) illustrates the throughput for different fault probabilities. In the practical indoor tests, the throughput of the proposed scheme is increased by 39.2% than BIM2RT, 38% than SAFT, and 37% than NLFFT. As predicted by the simulation results, the throughput is increased in the proposed scheme due to the efficient fault tolerance mechanism. Fig. 13(c) shows the network lifetime for different fault probabilities. In the indoor test, the network lifetime of the proposed scheme is increased by 64% than BIM2RT, 61% than SAFT, and 59% than NLFFT. The network lifetime of the proposed scheme is increased due to minimum message overhead and effective cluster formation. Fig. 13(d) shows the communication delay in different numbers of nodes. In a real hardware environment, the communication delay of the proposed scheme is reduced by 33% than BIM2RT, 31% than SAFT, and 30% than NLFFT. In the proposed scheme, the communication delay is reduced due to an adaptive routing mechanism based on the RLWOA algorithm. Fig. 13(e) depicts the average packet delivery in different numbers of nodes. In the practical scenario, the average packet delivery success rate of the proposed scheme is increased by 38% than BIM2RT, 36.4% than SAFT, and 35% than NLFFT. The hardware experiments show improvement in average packet delivery success rate which validates the results obtained through simulation. This improvement is due to the proposed optimal alternative path selection in the occurrence of link failure. Fig. 13(f) shows the recovery speed in the different number of rounds. The recovery speed of the proposed scheme is increased by 36.2% than BIM2RT, 35% than SAFT, and 34.1% than NLFFT. As described in the simulation result, the proposed hybrid RL-based fault tolerance mechanism helps to increase recovery speed.

VI. Conclusion

In this article, an intelligent fault tolerance scheme for WSN-assisted IIoT has been proposed. An intelligent fault-tolerant mechanism for node failure is presented which significantly improves the network lifetime. Furthermore, an intelligent fault-tolerant mechanism for link failure is proposed that significantly improves the network throughput and reduces communication delays. Extensive simulations have been performed to demonstrate the effectiveness of the proposed scheme in terms of average packet delivery, energy consumption, throughput, network lifetime, communication delay, and recovery speed. The proposed scheme is applied to develop a reliable fire detection system for industrial application. The proposed scheme can also be directly applied to other Industrial 4.0 applications such as process automation, industrial machine health monitoring, and emergency evacuation system. In future work, we plan to extend the proposed work by taking into account mobile sinks in the industrial environment.

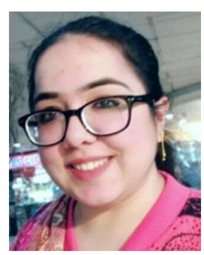
References

[1] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, “Energy efficient industrial Internet of Things: Overview and open issues,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 11, pp. 7225–7237, Nov. 2021.

[2] “Global smart machines in enterprise, industrial automation, and IIoT by technology, product, solution, and industry verticals 2022–2027,” 2022. Accessed: Apr. 15, 2022. [Online]. Available: https://www.researchandmarkets.com/r/njkgtg

[3] F. Al-Turjman and S. Alturjman, “Context-sensitive access in industrial Internet of Things (IIoT) healthcare applications,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 6, pp. 2736–2744, Jun. 2018.

[4] I. Lamrani, A. Banerjee, and S. K. S. Gupta, “Operational data-driven feedback for safety evaluation of agent-based cyber–physical systems,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 3367–3378, May 2021.
[5] X. Zhou et al., “Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1377–1386, Feb. 2021.
[6] “The growth of digitalisation with industry 4.0,” 2022. Accessed: Apr. 15, 2022. [Online]. Available: https://iottechnews.com/news/2022/feb/09/the-growth-of-digitalisation-with-industry-4-0/
[7] G. Kaur, P. Chanak, and M. Bhattacharya, “Energy efficient intelligent routing scheme for IoT-enabled WSNs,” *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11440–11449, Jul. 2021.
[8] H. Yang et al., “Fault-tolerant cooperative control of multiagent systems: A survey of trends and methodologies,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 4–17, Jan. 2020.
[9] G. Kaur, P. Chanak, and M. Bhattacharya, “Obstacle aware intelligent fault detection scheme for industrial wireless sensor networks,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 10, pp. 6876–6886, Oct. 2022.
[10] S. R. Pokhrel, S. Verma, S. Garg, A. K. Sharma, and J. Choi, “An efficient clustering framework for massive sensor networking in industrial Internet of Things,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4917–4924, Jul. 2021.
[11] A. K. Sangaiah et al., “Energy-aware geographic routing for real-time workforce monitoring in industrial informatics,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9753–9762, Jun. 2021.
[12] F. Ademaj and H.-P. Bernhard, “Quality-of-service-based minimal latency routing for wireless networks,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1811–1822, Mar. 2022.
[13] G. Künzel, L. S. Indrusiak, and C. E. Pereira, “Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5617–5625, Aug. 2020.
[14] X. Wang et al., “QoS and privacy-aware routing for 5G enabled industrial Internet of Things: A federated reinforcement learning approach,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, pp. 4189–4197, Jun. 2022.
[15] Y. Tong, L. Tian, L. Lin, and Z. Wang, “Fault tolerance mechanism combining static backup and dynamic timing monitoring for cluster heads,” *IEEE Access*, vol. 8, pp. 43277–43288, 2020.
[16] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. J. Jassbi, “Novel fault-tolerant clustering-based multipath algorithm (FTCM) for wireless sensor networks,” *Telecommun. Syst.*, vol. 74, no. 4, pp. 411–424, 2020.
[17] L. Rui, X. Wang, Y. Zhang, X. Wang, and X. Qiu, “A self-adaptive and fault-tolerant routing algorithm for wireless sensor networks in microgrids,” *Future Gener. Comput. Syst.*, vol. 100, pp. 35–45, 2019.
[18] T. Muhammed, R. Mehmood, A. Albeshri, and A. Alzahrani, “HCDSR: A. hierarchical clustered fault tolerant routing technique for iot-based smart societies,” in *Smart Infrastructure and Applications*. Cham, Switzerland: Springer, 2020, pp. 609–628.
[19] A.-R. Hedar, S. N. Abdulaziz, E. Mabrouk, and G. A. El-Sayed, “Wireless sensor networks fault-tolerance based on graph domination with parallel scatter search,” *Sensors*, vol. 20, no. 12, 2020, Art. no. 3509.
[20] H. Li, Q. Chen, Y. Ran, X. Niu, L. Chen, and H. Qin, “BIM2RT: Bwas-immune mechanism based multipath reliable transmission with fault tolerance in wireless sensor networks,” *Swarm Evol. Comput.*, vol. 47, pp. 44–55, 2019.
[21] V. K. Menaria, S. Jain, N. Raju, R. Kumari, A. Nayyar, and E. Hosain, “NLFFT: A novel fault tolerance model using artificial intelligence to improve performance in wireless sensor networks,” *IEEE Access*, vol. 8, pp. 149231–149254, 2020.
[22] J.-W. Lin, P. R. Chelliah, M.-C. Hsu, and J.-X. Hou, “Efficient fault-tolerant routing in IoT wireless sensor networks based on bipartite-flow graph modeling,” *IEEE Access*, vol. 7, pp. 14022–14034, 2019.
[23] P. Cong et al., “A deep reinforcement learning-based multi-optimality routing scheme for dynamic IoT networks,” *Comput. Netw.*, vol. 192, 2021, Art. no. 108057.
[24] T. Kegyes, Z. Süle, and J. Abonyi, “The applicability of reinforcement learning methods in the development of industry 4.0 applications,” *Complexity*, vol. 2021, pp. 1–30, 2021.
[25] G. Han, H. Wang, X. Miao, L. Liu, J. Jiang, and Y. Peng, “A dynamic multipath scheme for protecting source-location privacy using multiple sinks in WSNs intended for IIoT,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5527–5538, Aug. 2020.
[26] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.
[27] J. Duan, Z. Yi, D. Shi, C. Lin, X. Lu, and Z. Wang, “Reinforcement-learning-based optimal control of hybrid energy storage systems in hybrid AC–DC microgrids,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 9, pp. 5355–5364, Sep. 2019.
[28] W. Guo, C. Yan, and T. Lu, “Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing,” *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 2, 2019, Art. no. 1550147719833541.
[29] E. Emary, H. M. Zawbaa, and C. Grosan, “Experienced gray wolf optimization through reinforcement learning and neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 681–694, Mar. 2018.
[30] A. Seyyedabbasi, R. Aliyev, F. Kiani, M. U. Gulle, H. Basyildiz, and M. A. Shah, “Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems,” *Knowl.-Based Syst.*, vol. 223, 2021, Art. no. 107044.



Gagandeep Kaur (Graduate Student Member, IEEE) received the Ph.D. degree in computer science and engineering from the Atal Bihari Vajpayee-Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior, India, in 2022.

She is currently an Assistant Professor with Bennett University, Greater Noida, India. She has published various research articles in reputed international journals and conferences. Her research interests include wireless sensor networks, Internet of Things, soft computing, and machine learning.

Prasenjit Chanak (Member, IEEE) received the M.Tech and Ph.D. degrees in information technology from the Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India, in 2011 and 2016, respectively.

He is currently an Assistant Professor with Indian Institute of Technology (BHU), Varanasi, India. From December 2016 to August 2020, he also served as an Assistant Professor with the Department of Information and Communications Technology, Atal Bihari Vajpayee-Indian Institute of Information Technology and Management, Gwalior, India. He has executed various government of India-sponsored projects from DST, ICSSR, and SERB. He has authored or coauthored more than 60 research papers in different reputed international journals and conferences. He has published five Indian patents. His research interests are wireless sensor networks, Internet of Things (IoT), cyber-physical networks (CPN), machine learning, and consumer electronics systems.