IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 14, 2020, accepted August 3, 2020, date of publication August 12, 2020, date of current version August 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015985

NLFFT: A Novel Fault Tolerance Model Using Artificial Intelligence to Improve Performance in Wireless Sensor Networks

VINOD KUMAR MENARIA[1], S. C. JAIN[1], NAGA RAJU[2], RAJANI KUMARI[3], ANAND NAYYAR[4,6], (Senior Member, IEEE), AND EKLAS HOSAIN[5], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Rajasthan Technical University, Kota 324010, India
[2]Department of Computer Science, Central University of Rajasthan, Ajmer 305817, India
[3]Department of IT, JECRC University, Jaipur 303905, India
[4]Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[5]Department of Electrical Engineering and Renewable Energy, Oregon Tech, Klamath Falls, OR 97601, USA
[6]Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam

Corresponding authors: Anand Nayyar (anandnayyar@duytan.edu.vn) and Eklas Hosain (eklas.hossain@oit.edu)

ABSTRACT A wireless sensor network (WSN) is a collection of various tiny devices known as sensor nodes, which are also called motes. Due to high-energy consumption, the possibility of hardware, link or node failure, and some malicious attacks, sensor networks are considered error-prone networks. Hence, fault tolerance (FT) in WSN is one of the prominent issues. This article presents a novel FT approach named node-link failure fault tolerance model (NLFFT Model) in WSN, to handle the faults that occur either by link or node failure during data transmission from the sensor to the sink or base station. The NLFFT model consists of an improved quadratic minimum spanning tree (Imp-QMST) approach. This approach helps in finding the alternate link whenever it fails due to various situations and also an improved-handoff (Imp-Handoff) algorithm to support the node failure to the fault tolerance. Improved QMST presents a novel mechanism to find an alternate edge in place of the broken or failed edge in the spanning tree, to improve the fault tolerance in WSN. Imp-Handoff suggests a novel way to find the faulty node owing to less battery power and replaces a defective node by an appropriate neighbor to shift the tasks performed by a faulty node in WSN. Simulation results clearly state that as compared to the basic techniques i.e. Q-MST and Handoff algorithm, the proposed NLFFT model improvises the performance of WSN around by 7%. The results prove that the Imp-QMST gives about 6% improved throughput, 5% less end-to-end delay, and 6% less power consumption than the QMST algorithm. Similarly, Imp-Handoff improves about 4% throughput, 6% less end-to-end delay, and utilizes 7% less power consumption.

INDEX TERMS Fault tolerance, handoff mechanism, MST, Q-MST, swarm intelligence, WSN.

I. INTRODUCTION

A WSN comprises of many tiny devices known as sensor nodes and base stations. Such networks are useful for monitoring and passing environmental and physical constraints viz. temperature, pressure, and noise to a centrally located base station. The base station processes the data and concludes the necessary action. A typical WSN can be structured or unstructured, based on the locality where a sensor network needs to be established, to control the region with sensor devices. A structured network can be feasible in the region like buildings, streets, highways, and parking places, whereas unstructured networks can be viable in the forest, desert, flood situation, and disaster areas. In both kinds of networks, a large area is divided into clusters, and each cluster consists of tiny sensor devices (motes) and cluster head (CH). The sensor nodes in the cluster are connected by hop-to-hop with each other based on the coverage capacity among the existing sensors nodes and each cluster has a cluster header to communicate with other clusters. The clusters are interconnected through its cluster headers and finally aggregated data from all the clusters is transferred to the base station (BS), to process the information further or transfer to another heterogeneous network.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

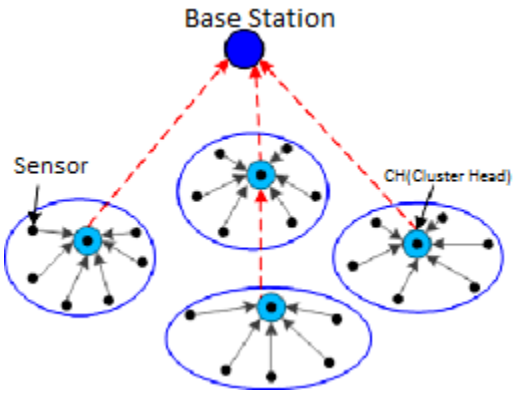

FIGURE 1. Typical WSN with various links.

Figure 1 shows an example of WSN with sensor nodes, CH, and BS. In the considered WSN, mainly four kinds of links exist namely sensor to sensor, sensor to CH, CH to CH, and CH to BS. The link existence between n_i to n_j is based on the available bandwidth $B(n_i, n_j)$ and active status of both nodes. Here, a node may be considered as sensor node, CH, BS. The link existence will exist based on the predefined threshold for various kinds of links. The proposed work applies to all kind of links mentioned in Fig. 1.

Based on the environment, sensor networks can be classified into various categories such as terrestrial WSN, underwater WSN, underground WSN, and mobile WSN. In some critical applications, such as critical-medical care [1], the sensor networks need to provide interminable services. Hence, enabling endless services in critical systems is only possible by fault tolerance. The efficient fault diagnosis method is required to find the faults to support fault tolerance. The fault tolerance is a combination of fault detection, fault diagnosis, and fault removal [2] processes.

WSNs are vulnerable to faults mainly because of two reasons- one is the failure of nodes, and the other is a link failure. Tiny and less battery power sensor nodes are deployed in a disastrous and harsh environment; hence, there may be higher chances of hardware failures in WSN. Due to faults in WSN, there may also be a loss in huge data transfer, long data transmission delay, and less throughput. So, there is a need for novel protocol architecture for WSN. Consequently, the identification of faulty nodes and links is a must to improve the capacity of WSN in terms of efficient data delivery.

Hence, a novel mechanism is needed to identify the faults and create a recovery mechanism to improve the reliability and quality of service during data transmission. The major contributions of the proposed research are as follows:

1. A novel fault-tolerance model is proposed to handle both kinds of faults i.e. link failure and node failure.
2. The Imp-QMST algorithm is introduced to find the alternate link in case of link failure.
3. The Imp-Handoff algorithm is presented to identify the faulty node and selection of appropriate new nodes in case of node failure.
4. Imp-QMST and Imp-Handoff need MST. Thus, four swarm intelligence-based algorithms have been used to generate spanning trees in WSN and the performance for different characteristics, i.e., throughput, end-to-end delay, and power dissipation of proposed methods to existing methods Q-MST [3-5] and Handoff [10] has been evaluated.

The rest of the paper is organized thus: Related work is explained in section II. Section III describes the proposed helping algorithms to find active links and nodes in the NLFFT model. Section IV narrates various system models and the newly proposed NLFFT model in detail. Section V elaborates on the proposed work along with examples and describes the various algorithms used in constructing MST to support the suggested fault tolerance model (given in section IV). Section VI explains the experimental setup. In section VII, the result analysis part highlights the experiments and performance comparison with other mechanisms namely PRIMS, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Imperialistic Competitive Algorithm (ICA) and Firefly (FF) algorithm concerning various network parameters viz. throughput, end-to-end delay, and power consumption. Finally, Section VIII concludes the paper highlighting the future scope.

II. RELATED WORK

This section presents a comprehensive literature survey of various routing protocols, data aggregation, and Fault Tolerance. Fault tolerance in WSN can be achieved broadly in two ways, one is to find the link failure, and the other one is to detect node failure. In the literature, there are several techniques available to find links and node failures individually. However, very few research techniques are available to find both node and link failures simultaneously. There are several techniques available in the literature to improve the reliability and quality of service (QoS) in WSN. The multipath [6] finding is the one to improve the reliability, timeliness, and load balancing in WSN. Faulty node detection and interference model caused by neighbor nodes are also equally popular in identifying the faults in WSN.

The work proposed in Lee and Choi [7] is a distributed algorithm to find the faults in WSN. The faulty nodes are identified based on the comparison of neighbor nodes and dissemination decisions made by each node. The proposed research identifies the transient faults in communication and sensing in WSN. A sliding window is utilized to eliminate the time redundancy.

Qu *et al.* [8] proposed a fault tolerance model with mobile agents to attain consistent and efficient performance, with required functions within a given period. The faults are detected and identified as the kind of failures owing to the behavior of the mobile agent. The behavior of the mobile agents can be analyzed statistically by various parameters viz. migration time from one sensor node to another sensor node, the lifetime of the mobile agents, and distribution of

mobile agent population. This mechanism overhead controls the behavior of mobile agents to adopt the changes in the environment.

Guo *et al.* [9] introduced the hybrid on-demand distance vector multi-path (HODVM) routing protocol for spatial wireless ad-hoc networks (SWAHN). The proposed protocol HODVM divides the SWAHN into two parts. The first part is with the backbone, and the second one is without a backbone. HODVM finds adaptive multipath to balance the workload and improve the fault tolerance in SWAHN. In WSN, maintaining multipath is a very complicated task due to the size and capacity of the links among various kinds of nodes.

Geeta *et al.* [10] described an active node based fault tolerance and interference (AFBTI) in sensor networks to pick out faulty nodes using two models, one being the battery power model, and another being interference model. They presented the handoff mechanism as a fault tolerance mechanism to face low battery power. However, the proposed work considers only battery power as criteria to decide the faulty node. It may not be appropriate in WSN.

Abba and Lee [11] proposed an autonomous self-aware and adaptive fault-tolerant routing technique (ASAART) for WSNs and compared the simulation results with self-selective routing and self-healing routing protocols in different simulation scenarios and found better in throughput, delay (end-to-end), packets error rate and power conservation in faulty and congested WSN. Yucai Zhou *et al.* [12] pointed out in their paper that due to faults or node failures occurring by abrupt environmental changes, sensor network performance gets deteriorated. Hence, to solve the problem mentioned above, a protocol supporting high fault tolerance and power efficiency based on a multi-way routing mechanism has been proposed. It is an enhancement in hybrid, energy-efficient distributed clustering mechanism (HEED) protocol, called HEED-FT. The HEED-FT mechanism consumes less energy and provides high reliability and increases network lifetime.

Sutagundar *et al.* [13] proposed a fault-tolerance approach based on multiple (both fixed and mobile) agents. This proposed multi-agent-based scheme provides fault tolerance at the node, cluster, and sinks level. In node-level fault tolerance, a node accepts only correct samples i.e., the samples in the reference range and rejects the false samples. Thus, the node calculates the average of all samples which were found correct. At cluster level fault tolerance, after receiving the data from all the nodes, the difference in data is computed by comparing the data of each node to all other nodes. Finally, fault tolerance at the sink level, the sink node broadcasts alive packets to every node and replicates the sink node periodically. If nodes and replicated sink nodes get alive packets from the sink node, then it transmits the data to the sink otherwise replicated sink node broadcasts its ID to every node.

Tien *et al.* [14] proposed dual separate paths (DSP) algorithm to support fault-tolerance by improving network traffic in WSNs. The DSP algorithm constructs two different paths between the source and target (destination) nodes, depending on the topology of the network. This algorithm supports both wired and wireless networks and provides non-interminable fault tolerance. Elsayed *et al.* [2] presented an approach called distributed self-healing approach abbreviated as DSHA, wherein fault identification, diagnosis, and repairs are performed. These are skilled at both the cluster head and node level. DSHA mechanism was found effective in identifying hardware failures and diagnoses to make sensor networks resilient and reliable. The results showed that the DSHA approach can handle up to 67.3% hardware malfunctioning and hence improve 62.6% network lifetime.

Mitra and Das [15] observed that network reliability and dependability can be measured by its fault identification, fault diagnosis, and overcome techniques. The proposed architecture supported distributed fault tolerance and algorithms for fault recovery by using checkpoints of data and state of nodes in a distributed environment. Ma *et al.* [16] developed a novel algorithm for fault tolerance by generating a multi-routing tree abbreviated as FTMRT. In it, a multi-routing tree is constructed to ensure k-disjoints routes from every sensor node to a collection of super-nodes. Thus, to build a fault-tolerant topology, each data transmitting node redefines the transmission power in compliance with the multi-routing tree.

Begum and Nandury [17], proposed a fault-tolerant mechanism called component-based self-healing mechanism which builds an alternate path to reach the root node in a hierarchical aggregation tree when any node or link fails. The said mechanism is based on two algorithms, the first is Self-healing Component-based Reconfiguration (SCR) for preserving precedence relations and another one is SCR-Dynamic Transmission Range Adjustment (SCR-DTRA). These two approaches are compared with existing approaches i.e. Tree Reconstruction, WKLP, and FTS. Simulation results prove that the SCR-DTRA approach is effective in terms of the Affected Ratio (AR) and Recovery Ratio (RR).

In [18], Kim *et al.* introduced a Maximum fault tolerance barrier coverage problem related to hybrid sensor networks (MFBP-HSN), containing both static and fully-controllable dynamic (mobile) sensors. It tried to migrate the mobile nodes in such a manner that fault tolerance of barrier-coverage of hybrid sensor networks mentioned above, was maximized. A polynomial-time exact algorithm was proposed for the above-said problem.

Shagufta Henna [19] presented an algorithm for approximation known as energy-efficient maximum disjoint coverage (EMDC) with a resemblance estimation ratio. In the proposed work, EMDC performance analyses showed up in favor of power efficiency and fault tolerance. The EMDC algorithm increased the lifespan of the network by choosing two disarticulated set covers minimizing relay power. The EMDC mechanism was found better in terms of network lifetime in comparison of DSC-MDC for a different count of nodes and ranges.

Moussa and Elalaoui [20] proposed a mechanism related to fault tolerance of cluster heads. The simulation was performed and found better as compared to low energy adaptive clustering protocol (LEACH), and informer homed routing mechanism.

To overcome the challenges of WSN, Jassbi and Moridi [21] presented an algorithm called the HEED algorithm for clustering. Further, for improving power consumption, they proposed a sleep/wake-up method for the cluster members. The main focus of work [21] was to identify and repair faults of cluster heads as well as cluster member sensor nodes. Hence, to perform this task, a node is chosen as a backup node of the cluster head. The failed node is separated, and its place is taken by its neighbor node, which goes to wake-up mode from the sleep mode.

Hence, by anatomizing the literature it has been found that a unique mechanism is needed to diagnose the faults and novel methods must be discovered to ameliorate the reliability and quality of service during the transmission of data. Further, the existing literature on spanning trees to achieve data aggregation is devoid of any fault tolerance procedure and thus, it influences the various important parameters of WSN viz., throughput, end-to-end delay, and power consumption.

III. PROCEDURES TO FIND ACTIVE NODES AND ACTIVE LINKS

In WSN, fault tolerance is a crucial issue, and it can be achieved by finding the active node and active link. The number of active nodes and links, define the fault tolerance factor of WSN. In this proposed model, we derived a procedure to find the active node and link. The proposed dynamic node model makes use of the neighbor node, average battery power, and distance. The dynamic connection can be identified based on the available bandwidth over the link. The following sections illustrate the mechanism to find the active node and link separately.

A. ACTIVE NODE IN WSN USING BATTERY POWER AND DISTANCE

The active state of the node is represented by $n_{active} = [n_{bp}, d]$, where n_{bp} is the battery power of node, and d is the distance between the intermediate node and its neighbor node. In the proposed model, the active node is decided based on the heuristic method, in which existing neighbor node battery power and distance are compared with predefined threshold values. The threshold values are node's average battery power denoted by n_{avg-bp} and average distance indicated by d_{avg}. These values are evaluated based on the existing neighbor node's average battery power and average distance. The following is the procedure to find the active node in the improved handoff algorithm (refer algorithm 1).

B. ACTIVE LINK IN WSN USING AVAILABLE BANDWIDTH

In the given WSN, all sensor nodes are connected with the base station, cluster head (CH), and with other sensor nodes with specific bandwidth (i.e. frequency range). In the proposed model, certain bandwidth thresholds are defined between sensor to sensor (B_{s-s}), a sensor to CH (B_{s-ch}), CH to CH (B_{ch-ch}), and CH to Base Station (B_{ch-bs}). If any link (edge) weight is less than the threshold, then it is considered as a broken link. The thresholds among the various kinds of links and bandwidth thresholds are given in Table 1.

The following procedure is used to find the active link in the improved Q-MST algorithm (refer algorithm 2).

Algorithm 1 Procedure to Find Active Node

```
Procedure (Node, BP, Dist)
Begin:
  If (BP > n_avg-bp && Dist > d_avg)
   Then
     Return active
  Else
     Return Inactive
End
```

TABLE 1. Various thresholds.

S. No	Type of the Link	Bandwidth Threshold
1	Sensor to Sensor	B_{s-s}
2	Sensor to Cluster Head	B_{s-ch}
3	Cluster Head to Cluster Head	B_{ch-ch}
4	Cluster Head to Base Station	B_{ch-bs}

IV. PROPOSED SYSTEM MODELS

Various system models used in this article are discussed in the subsequent section:

A. PATH LOSS MODEL

In wireless communication literature, various path loss models and their analysis exist. The followings are well-known path loss models in wireless communication.

- Free-space model
- Two-ray model
- The simplified path loss model
- Empirical models.

Path loss is defined as $P_L = (P_t/P_r) > 1$ where, P_L is the path loss, P_t is power transmission, and P_r is received power.

$$P_L(dB) = 10\log_{10}(P_t/P_r) > 0 \quad (1)$$

The radio signals, as propagates away from the transmitters, the power of the radio signals is reduced according to distance traveled from the transmitters. In most of the wireless communication systems, often, the default path loss model is the free space path loss model, which computes attenuations according to the inverse square law along a single line-of-sight propagation path.

In all of the above-said models, the path loss is proportional to distance with some exponent, and exponent depends on the path model. Further, in all of the path loss models, the distance from the receiver and transmitter is the key factor. Hence, in our proposed model, we have considered the average distance, which is the threshold to decide the active node.

Algorithm 2 Procedure for Finding Active Link

```
Procedure (Type of the Link, Bandwidth)
Begin:
  If (Type == 1)
    If (Bandwidth > B_{s-s})
    Then
      Return active.
    Else
      Return Inactive.
  Else
    If (Type == 2)
      If (Bandwidth > B_{s-ch})
      Then
        Return active
      Else
        Return Inactive
  Else
    If (Type == 3)
      If (Bandwidth > B_{ch-ch})
      Then
        Return active
      Else
        Return Inactive
  Else
    If (Type == 4)
      If (Bandwidth > B_{ch-bs})
      Then
        Return Active
      Else
        Return Inactive
  END:
END:
```

Figure 2 shows the path loss in the first order radio model and its relation with distance. In WSN currently, there is a great deal of low radio energy consumption models. In the proposed model, we have adopted first order radio model for simplicity and the radio dissipates $E_{elec} = 50$ nJ/bit to run the transmitter or receiver circuitry and $\mathcal{E}_{\text{amp}} = 0.1$ nJ/bit/m^2 for the transmit amplifier. These parameters also are shown in Table 5.

$$P_L = P_t/P_r \; \alpha \; d^n \tag{2}$$

The eq. (2) indicates the path loss model, which depends on the radio transmission and receiving power, which is proportional to some of the power of the distance in all the well-known path loss models [22]. The distance between two nodes (x_i, y_i) and (x_j, y_j) will be evaluated using the formula $sqrt[(x_j - x_i)^2 + (y_j - y_i)^2]$. P_L is path loss from each neighbor at node n from each neighbor node 1, 2, 3, 4, and 5.

In the proposed model, using the average distance (d_{avg}) among n nodes, consider nodes 1, 3, and 5 are active and thus deemed to intended. Similarly, nodes 2 and 4 are inactive; hence, these are not considered to intend. In nodes 2 and 4, the interference problem arises, and it is handled

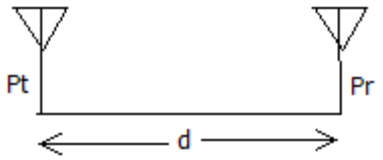

FIGURE 2. Path Loss Model.

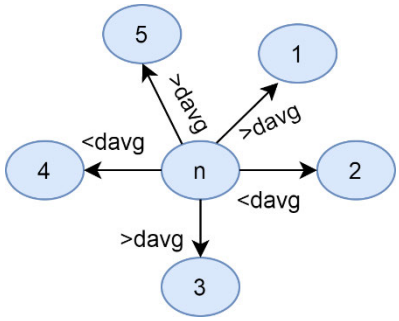

FIGURE 3. Active Neighbor Node based on Average Distance Due to Interference.

by the interference model given in [10], but according to suggested work in this article, it is essential not to consider the interference model as already selecting the node of a higher average distance. Further, due to interference, there is a kind of node or link failure that has been previously handled by the proposed fault tolerance model; hence, the interference model is not required in the proposed approach.

B. THE FLOW CHART OF THE PROPOSED NLF FAULT TOLERANCE MODEL

A model for fault tolerance is proposed to support both node and link failure; its functioning has been presented here in Fig. 4. For a link or edge failure, the Imp-QMST algorithm and, for node failure, the Imp-Handoff algorithm have been proposed, which are improved forms of QMST and Handoff algorithm, respectively. A WSN comprises of several sensor nodes. Let an undirected fully connected graph G = (V, E) represent the topology of the network where $V = \{v_1, v_2, v_3, \ldots . v_n\}$, stands for a set of nodes, $E = \{(v_i, v_j) | v_i, v_j \in V \wedge d(v_i, v_j) < \min(R_i, R_j)\}$ are the set of edges and $d(v_i, v_j)$ is Euclidean distance of v_i, v_j and R_i is the transmission range of v_i.

The proposed NLFFT model control flow illustrated in Fig. 4.

C. ENERGY CONSUMPTION MODEL OF WSN

A variety of power consumption models [23] are available, but this article uses Heinzelman *et al.* [24] model for power dissipation and the meaning of symbols has been illustrated in Table 5. The proposed work uses, to transmit and receive an n-bit message to a distance (d) by using radio model:

Energy Consumption at Transmitter (refer Eq. 3)

$$E_{Tx} = E_{Tx-ele} \times n + \varepsilon_{amp} \times d^2 \tag{3}$$

Energy Consumption at Receiver (refer Eq. 4)

$$E_{Rx} = E_{Rx-ele} \times n \tag{4}$$

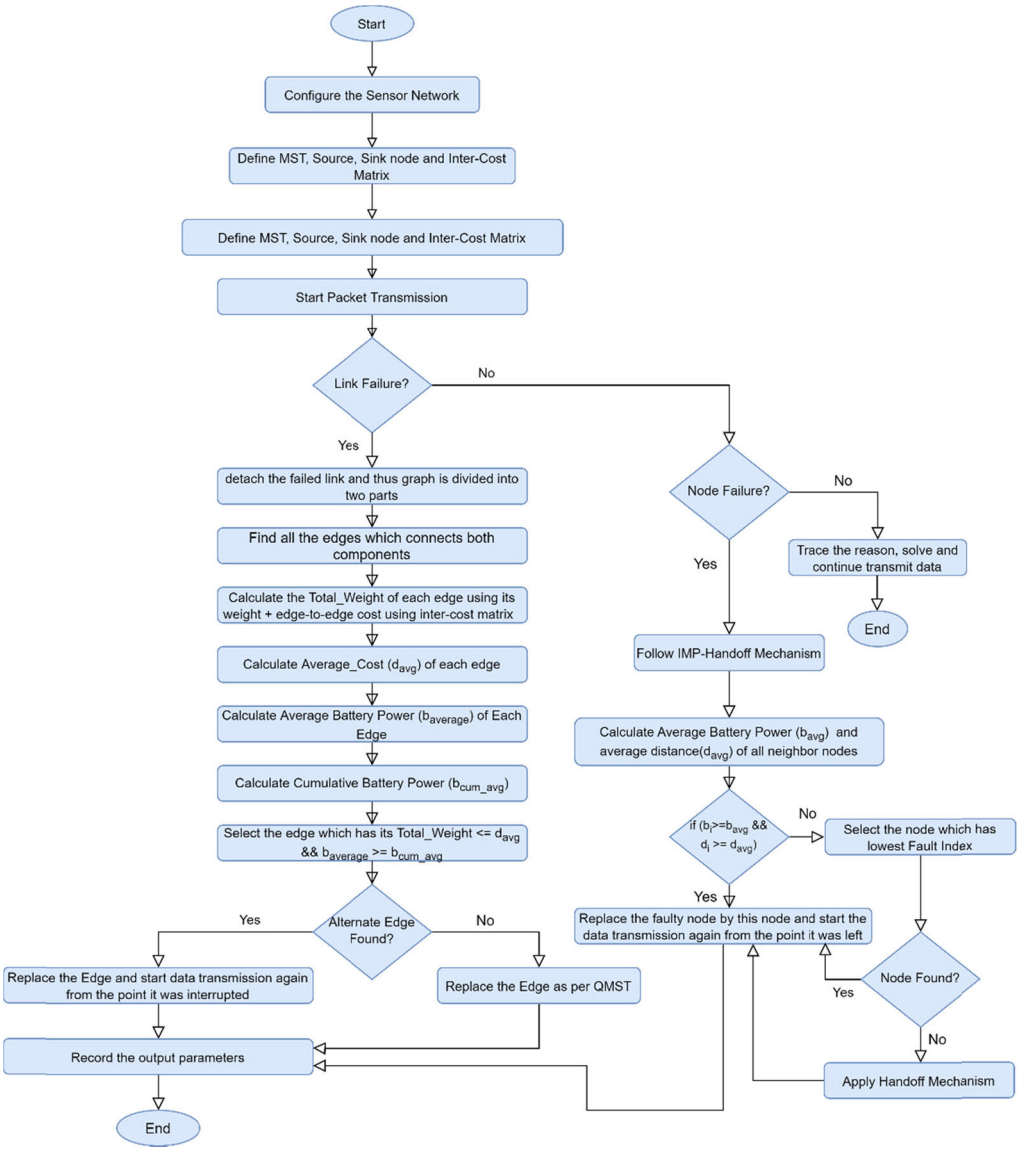

FIGURE 4. Proposed NLFFT Model.

In the simulation, it summarizes that the size of every incoming and outgoing message is one bit. Hence, the Eq. (2) and Eq. (3) are converted into (4) and (5) respectively.

$$E_{Tx} = E_{Tx-ele} + \varepsilon_{amp} \times d^2 \tag{5}$$

$$E_{Rx} = E_{Rx-ele} \tag{6}$$

D. END TO END DELAY MODEL

To calculate the end-to-end delay, the following formulae have been used:

The Time required to send a packet at k-links

$$= k^*[(H + (D/P))/C] \tag{7}$$

Here, H is the header size (in bits), D is bit size of data, P is some packets and C is the link capacity, also known as buffer size. It has been assumed that the header size (H) of each packet is 2 bits and the data size is (D) 1 bit. All the links are considered as homogeneous and having a capacity to store 50 packets (identical to the link capacity).

Thus, it can be seen that the number of packets is identical to the sum of the size of all the data packets. Thus, by putting D/P = 1, H = 2, and C = 5 the above-given formula (refer Eq. 7) is converted to Eq. (8).

$$End\_to\_End\_Delay = (3/5) \times k \tag{8}$$

V. NLFFT MODEL: A NOVEL FAULT TOLERANCE MODEL TO SUPPORT FAILURE AT BOTH LINK AND NODE LEVEL

Though the flow chart of the NLLFT model has been presented in the previous section, this section highlights the implementation of the techniques used in the proposed NLFFT model.

Firstly, MST is generated by using swarm-based approaches (ACO, PSO, FF, and ICA algorithm) and traditional PRIMS algorithm to implement the proposed NLFFT

model for fault tolerance. Every node as a part of MST acts as a data aggregator node. The Prims algorithm is known as Jarnik's algorithm, which is a greedy algorithm to find MST by calculating the subset of edges that construct a tree, including all vertices, at least once.

Marco Dorigo gave the ant colony optimization (ACO) [25] in 2006. In ACO, artificial ants are deployed to perform a heuristic search. The ACO [26]–[28] algorithm uses a probabilistic approach to get MST, which acts as a data aggregation tree. After obtaining the desired solution, the pheromone and evaporation parameters get updated.

The PSO algorithm is a meta-heuristic and population-based algorithm. It was developed by Eberhart and Kennedy in the year 1995 [29]. It is a nature-inspired and swarm-intelligence based heuristics method, provided by recognizing the movement and behavior of birds and fish flocks. The minimum spanning tree using the PSO algorithm and various parameters related to it have been explained by Goldbarg *et al.* [30] and improved by various researchers for performance improvement [31].

The Firefly algorithm was developed by Xin-She Yang at the end of the year 2007 and the beginning of 2008 [32], [33], similar to PSO, it is also a meta-heuristic algorithm inspired by the flashing behavior of fireflies and their bioluminescent communication [34]. A discrete Firefly algorithm for improved data collection and data aggregation in WSN was elaborated in [35], [36]. The ICA [37] algorithm was given by Atashpaz-Gargari and Lucas in 2007[38]. ICA can be considered as a human counterpart of the Genetic Algorithm [39]. It performs human social evolution while the Genetic Algorithm performs biological evolution of species. Sayadnavard *et al.* explained sensor network localization using the ICA algorithm [40].

The ABC algorithm was presented by Devis Karaboga [41] in the year 2005. It is a meta-heuristic algorithm based on the swarm. It was designed to optimize and solve multivariable and numerical problems [42], and it was influenced by the honey bees and their genius foraging behavior. The ABC algorithm contains three components, mainly, i.e., working bees, unemployed bees, and the food source. The detailed algorithm related to this article, to solve Q-MST using ABC, is explained in [4]. In the NLFFT model, an improved version of QMST [3]–[5] and Improved Handoff algorithm were proposed and named as Imp-QMST and Imp-Handoff algorithms respectively. Tools generally used for WSN are discussed in [43] and energy-efficient protocols are developed in [44]. A detailed study of the swarm and evolutionary algorithm available in [45]–[49].

A. IMP-QMST MECHANISM

In the Q-MST approach, at the place of edge cost (weight), the cost of enjoined pair of edges is also considered, and the target is to find out the side with the lowest side cost, to be replaced at the place of broken link or edge. The Q-MST is an NP-Hard problem, which was suggested by Asad and Xu [42]. Further, it was extended by Sundar and Singh [4] by implementing an ABC algorithm to find MST. The concept of applying fault tolerance in WSN using the Q-MST approach, along with an example, has been implemented by Menaria *et al.* [3].

Going a step ahead of the above-mentioned Q-MST approach, this article proposes an improved-QMST (Imp-QMST) mechanism. The flowchart of Imp-QMST has been shown in section 4.A.2 and can be explained thus:

1. When a node or link breaks down, or failure occurs due to battery power or any other reason, the sensor network is separated into two components. (one remains to the left side of the broken link and another to the right side)
2. Find the edges which connect the two parts mentioned above (sets).
3. Calculate the total cost (*Total_Weight*) of every connecting edge mentioned in step 2, by using the inter-cost matrix as procedure mentioned in Q-MST [4], i.e., the cost (weight) from each existing edge to every connecting edge of above components and vice versa. (Detailed example provided in [3])
4. Now, instead of selecting minimum cost edge (Q-MST), calculate the average cost d_{avg}, of all the corresponding costs of sides calculated in step 3.
5. Calculate average battery power ($b_{average}$) for each edge separately using Eq. 9:

$$b_{average} = \frac{\sum_{i=1}^{2} bi}{2} \tag{9}$$

   where b_i is the battery power of i^{th} node on the edge e_i.
6. Similarly, calculate the cumulative average battery power ($B_{cum-avg}$) for all the edges using Eq. 10, which are included in the connecting sets, i.e. edges which connect both components.

$$B_{cum-avg} = \frac{\sum_{i=1}^{n} b_{average}(i)}{n} \tag{10}$$

7. Edge Selection (Select the edge which has its cost lower than or equals to average cost d_{avg}, and its average battery power $b_{average}$ is higher than or equal to cumulative average battery power i.e. $B_{cum-avg}$.
8. If no edge is found in step 6, then follow a simple Q-MST mechanism, i.e., select the edge with minimum cost as calculated by step 3.
9. Replace the selected edge at the broken edge or link in the network and continue transmissions.
10. End

Let's consider the steps mentioned above with a sensor network taken as an example in Fig. 5.

Figure 5(a) shows a complete undirected graph with 5 sensor nodes, and these nodes are connected through 10 numbers of sides (edges), called links in the WSN ranging from e1 to e10. The inter-cost matrix (edge-to-edge cost matrix) of the said network has been shown in Table 2.

Table 3 depicts the battery power at the given moment, for all the five nodes participating in the network.

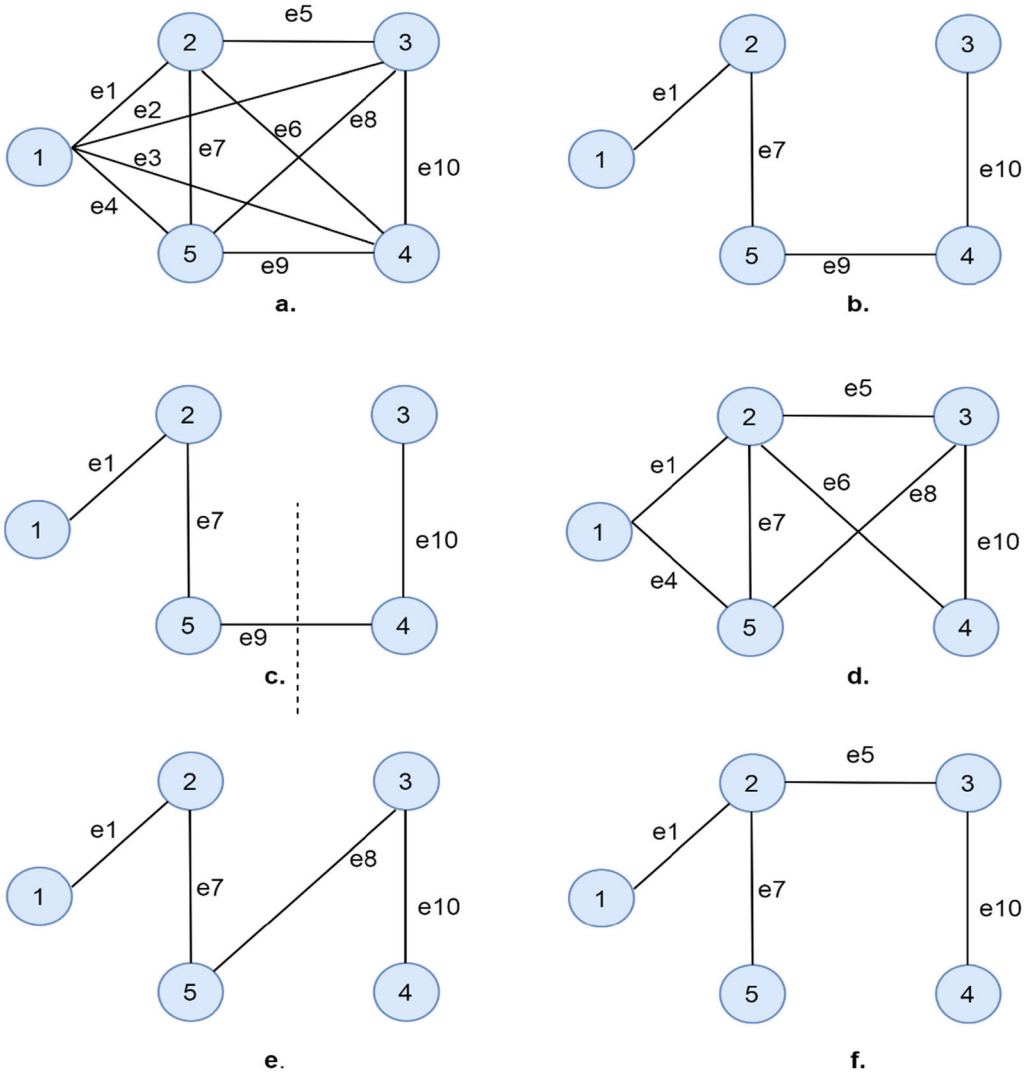

FIGURE 5. A Sensor network with QMST/Imp-QMST Process.

TABLE 2. Inter-cost matrix.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
e1	52	12	17	18	8	9	7	10	14	22
e2	9	94	14	18	15	14	11	10	5	12
e3	10	7	82	21	16	19	17	12	16	7
e4	13	21	8	62	9	14	12	15	10	11
e5	7	10	8	8	76	9	12	20	16	8
e6	7	6	10	17	3	80	11	21	19	7
e7	15	7	20	12	20	14	67	9	22	18
e8	5	21	6	8	12	19	16	87	11	15
e9	3	12	14	13	15	11	10	8	47	13
e10	11	10	8	20	14	12	19	8	5	57

In the mentioned inter-cost matrix in Table 2, each diagonal element presents the edge weights (cost or distance), and the remaining elements are edge-to-edge costs, which can be considered as a parameter related to WSN, i.e., channel

TABLE 3. Battery Power of all the nodes during link failure.

NODE	1	2	3	4	5
BATTERY POWER	383.50	250.50	700.35	405.62	320.20

strength. Figure 5(b) is a minimum spanning tree with a total weight of 223. Now suppose, due to any valid reason, the link e9 becomes weak or broken down (refer Fig. 5(c)), then the whole graph is separated into two sets, one is {1, 2, 5}, and another set is {3, 4}.

Now, it is required to select the alternate edge or side connecting the said two sets. Fig. 5(d) depicts that there are 3 possible edges connecting the two sets namely {e_5, e6, e8). Now, the total weight (cost) is to be calculated for all three edges. First, an edge-to-edge cost is calculated by considering every possible alternate edge using following steps:

- For the link e5 (connecting the vertexes 2 and 3)

$$\text{Total Edge Cost} = \text{cost}(e_5, e_5) + \text{cost}(e_1, e_5)$$

$$
\begin{aligned}
&+\text{cost}(e_5, e_1)\\
&+\text{cost}(e_7, e_5) + \text{cost}\,(e_5, e_7)\\
&+\text{cost}\,(e_{10}, e_5) + \text{cost}\,(e_5, e_{10});\\
&= 76 + 8 + 7 + 20 + 12 + 14 + 8\\
&= 145
\end{aligned}
$$

- For the link e6 (connecting the vertexes 2 and 3)

$$
\begin{aligned}
\text{Total Edge Cost} &= \text{cost}\,(e_6, e_6) + \text{cost}(e_1, e_6)\\
&+\text{cost}\,(e_6, e_1)\\
&+\text{cost}(e_7, e_6)\\
&+\text{cost}\,(e_6, e_7) + \text{cost}\,(e_{10}, e_6)\\
&+\text{cost}\,(e_6, e_{10})\\
&= 80 + 9 + 7 + 14 + 11 + 12 + 7\\
&= 140
\end{aligned}
$$

- For the link e8 (connecting the vertexes 2 and 3)

$$
\begin{aligned}
\text{Total Edge Cost} &= \text{cost}\,(e_8, e_8) + \text{cost}(e_1, e_8)\\
&+\text{cost}\,(e_8, e_1) + \text{cost}(e_7, e_8)\\
&+\text{cost}\,(e_8, e_7) + \text{cost}(e_{10}, e_8)\\
&+\text{cost}\,(e_8, e_{10})\\
&= 87 + 10 + 5 + 9 + 16 + 8 + 15\\
&= 150
\end{aligned}
$$

As the total weight (cost) of edge e6 is low i.e., 140 than other replaceable edges hence, in QMST approach the broken edge, i.e. e9 will be replaced by e_6 depicted in Fig. 5(e) but, Imp-QMST will apply some modifications and other parameters viz. battery power as:

1. Calculate the average weight (cost) of all the alternate edges

$$d_{avg} = (147 + 140 + 150)/3$$

2. Calculate the average battery power ($b_{average}$) of each edge connecting both the sets (SET-1 and SET-2) as:

a) $b_{average}$ of edge e_5 = (250.50+700.35)/2.0
= 475.42

b) $b_{average}$ of edge e_6 = (250.50+405.62)/2.0
= 328.06

c) $b_{average}$ of edge e_8 = (320.20+700.35)/2.0
= 510.27

3. Calculate the cumulative average battery power ($b_{cum-avg}$) of all the nodes participating in connecting the two sets:

$$
\begin{aligned}
b_{cum-avg} &= (475.42 + 328.06 + 510.27)/3;\\
&= 437.91
\end{aligned}
$$

4. Now, the selection process starts by using the information provided in steps 1 to 3 and can be concluded by making a decision table, i.e. Table 4.

Thus, it can be seen clearly in the decision table that the edge e_5, having the link cost lower than average cost,

TABLE 4. Decision table.

edge	$Edge\_Cost <= d_{avg}$	$b_{average} >= b_{cum-avg}$
e_5	YES	YES
e_6	YES	NO
e_8	NO	YES

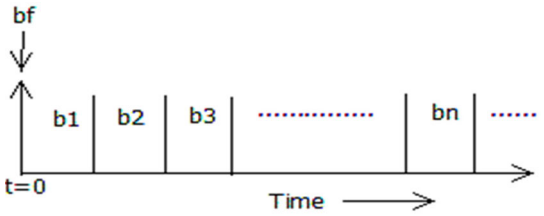

FIGURE 6. Time Frame for Battery Energy Level.

i.e.145<=145.66 and its average battery power is higher than the cumulative battery power, i.e., 475.42>=437.91). Hence, the edge e_5 will get selected as an alternative edge at the place of broken edge e9, as shown in Fig. 5(f).

The side e6 and e8 do not fulfill both the conditions, so these sides will not be considered. Now the question arises that by following the above Imp-QMST procedure, what to do when more than one edge found suitable. In this case, among all suitable edges, the edge with minimum cost gets selected as a replaceable edge.

B. IMP-HANDOFF MECHANISM

Similar to the QMST mechanism, the Handoff algorithm is used in fault tolerance to handle node failure in WSN. The only difference is that Q-MST or proposed Imp-QMST algorithm works on a link or edge failure while the Handoff or Imp-Handoff mechanism works on node failure. The flowchart of the Imp-Handoff mechanism is shown in section 4.A.2 in the proposed NLF Fault Tolerance Model. The algorithm is shown in this section.

The handoff mechanism [10] is used to replace the faulty node with an alternate sensor node. In the handoff mechanism, whenever a fault occurs at any node due to low battery power or power failure, the neighboring node with the highest battery power replaces the faulty node, and all services running on the defective node get transferred to the new neighbor node. Thus, the data transmission process does not get interrupted.

The Imp-Handoff algorithm is applied as the fault tolerance technique because of the battery power drain mentioned in Fig. 6.

Whenever a node, either a malicious or non-faulty, recognizes that its battery energy level is reduced to a degree up to b_{th}, i.e., if $b_k \leq b_{th}$ (b_k indicate to the k^{th} time window), a handoff connection to its neighbor node needs to be initiated. In the Handoff mechanism, the malicious node gathers the status of the energy level of all its neighbor nodes and transmits the handoff parameters to the neighbor node, containing the superior battery power. The battery power gathering process has two phases: (1) sending a request of

battery power to all neighbor nodes (2) all the neighbors, send reply packets having their latest battery power.

In the Imp-handoff mechanism, besides including the process mentioned in the above paragraph (related to the battery power), two more parameters are also considered- namely the distance (weight) of all neighbor nodes to faulty node and fault index of every node. The term fault index (FI) is defined as:

$$FI = \frac{n_{missed}}{n_{total}} \tag{11}$$

where n_{missed} = total number of missed packets and n_{total} is the total number of packets.

Initially, the fault index of every node is considered zero, but as soon as transmission starts, its value gets updated. After receiving all the parameters, the average battery power and average distance for all the neighbor nodes are calculated.

At first, the neighbor node with battery power higher than or equals to average battery power and with distance higher than or equal to the average distance will get selected. While in some cases, if the above condition does not get satisfied, then the neighbor node with the least fault index will get selected. The notations used and the Imp-Handoff algorithm are presented below:

Algorithm 3 Improved Handoff Algorithm (Imp-Handoff Algorithm)

1. Initially assign a battery power (b_i) to every sensor node arbitrarily.
2. Define Source, Sink, and Threshold value (b_{th}) of the Battery Power.
3. Start data transmission from defined source to destination repeatedly.
4. Update the battery power for every transmission defined in step 3.
5. IF any fault occurs in step 3 ($b_k < b_{th}$) then
6. Send REQUEST packets to all neighbor nodes for both distance and updated battery power excluding next-hop.
7. Receive the REPLY packets in the response of request given in step 6, from each n-1 neighbor node.
8. Arrange all the neighbor nodes of a failed node in ascending order of distance.
9. Calculate the parameters $b_{average}$ and d_{avg}.
10. For each node from the list of (n-1) nodes
11. Check whether it's the last updated battery power $\geq b_{average}$ and distance $\geq d_{avg}$.
12. If any suitable node found in step 10 then replace the faulty node from this node by sending all parameters related to connection to the identified node i.
13. Else select the node which has the lowest fault index (FI_i).
14. Replace the faulty node with the identified node in steps 8 to 12.
15. End

Notations: b_k is the battery power of the faulty node in a k^{th} time frame, b_{th} is the threshold power which is a predefined constant, n is the count of neighbor nodes, d_k is the distance (weight) of a neighbor node from the faulty node, $b_{average}$ is the average battery power, d_{avg} is the average distance (weight) of neighbors from faulty nodes, and FI_i is the fault index of node i.

The above proposed Imp-Handoff algorithm is compared with the existing handoff algorithm. The results have been analyzed in section 7, with the experimental setup discussed in section 6.

VI. EXPERIMENTAL SETUP

A. ASSUMPTIONS

While examining the performance of the proposed NLFFT model, the following assumptions are considered:

- It is assumed that every node in the sensor network setup is homogeneous and is deployed statically.
- Every sensor node has an equal capability to sense the surrounding data.
- The probabilistic approach is taken into consideration from the Prowler Simulator for networks, in the context of packet transmission from source to sink node. In this approach, every packet has a probability between 0 and 1 and those packets that have a probability greater than 0.5, will only be transmitted to the next hop.
- After constructing the MST using various swarm intelligence techniques, this MST is considered as a data aggregation tree where all the leaf nodes are considered as data aggregator nodes.

B. SIMULATIONS AND SIMULATION PARAMETERS

The simulation was performed in the MATLAB environment. The sensor nodes were set out randomly in a 200∗200 square unit area. The source and destination (sink) nodes were also declared arbitrary. The Euclidean distance of all the pairs of connected nodes was calculated and if it was found less than 150 meters, then those nodes could be considered as disconnected. Finally, the depth-first search (DFS) was imposed on examining whether the network was connected or disconnected. If the network was found as disconnected, then the whole procedure was repeated to get the connected graph to carry out the further simulation.

After getting a connected graph, MST is originated with various algorithms, as mentioned in section V, and then performs packet transmission. While carrying out packet transmission, if any node or link fails due to less battery power or power failure, Handoff and Imp-Handoff algorithms, Q-MST, and Imp-QMST algorithms are applied to support fault tolerance, as elaborated in section V.

As the packet transmission process is completed, various parameters like the total number of packets transferred, total packets received, total packets lost, the power consumed, and the end-to-end delay are calculated. The various comparative graphs are plotted in the result analysis in section VII.

TABLE 5. Simulation parameters.

SYMBOL	MEANING	VALUES
N	Number of Sensor nodes	10-60
S_A	Simulation Area	200×200
R	Transmission range	150 m
E_{Tx-ele} /E_{Rx-ele}	Radio Dissipation	50 nJ/bit
ϵ_{amp}	Transmit amplifier	0.1 nJ/bit/m^2
ρ	Pheromone evaporation parameter	0.3
γ	The relative influence of heuristic values η(i,j)	20
ω	Inertia Weight	0.2
ω_{damp}	Inertia Weight Damping Ratio	1.0
C1	Personal Learning Coefficient	0.7
C2	Global Learning Coefficient	1.0
μ	mutation rate	0.1
P_{revol}	Revolution Probability	0.5

Table 5 enlists various simulations parameters that are considered as an experimental setup to measure the performance of the proposed NLFFT model.

VII. RESULT ANALYSIS

Competitive analysis between QMST versus Imp-QMST and Handoff versus Imp-Handoff has been performed through various swarm intelligence techniques viz. ACO, PSO, FF, ICA, and traditional PRIMS algorithms.

Initially, MST is constructed through techniques as mentioned above. Whenever there are link or node failure, the said fault tolerance techniques, i.e. Q-MST, Imp-QMST, Handoff, and Imp-Handoff are applied. Results are calculated in terms of throughput, end-to-end delay, and energy consumed during the whole data transmission process. After performing several simulations, for each of the transmission, the average results are computed, as shown in Table 5. Results for QMST and Imp-QMST have been discussed in section VIII part A.

A. QMST VS IMP-QMST ALGORITHM

In this section, MST is generated by using the above-mentioned swarm intelligence algorithms and the PRIMs algorithm. Further, QMST and Imp-QMST are applied for the fault-tolerance purpose. The QMST and Imp-QMST algorithms deploy the ABC algorithm to find an alternate edge or link at the place of the broken link. The comparison of results has been depicted in subsequent sections.

1) THROUGHPUT COMPARISON

Table 6 and Fig. 7 shows the throughput comparison between QMST and the Imp-QMST approach. The graphical presentation of throughput comparison is shown in Fig. 7. It is clear by analyzing Table 6 and Fig. 7 that the throughput received through Imp-QMST is almost 2% better than the Q-MST approach. If it is analyzed algorithm wise, then it is found that Imp-QMST is 7% better than QMST for the PRIMS algorithm, 3% for ACO algorithm, 7% better for PSO algorithm, 6% better for FF algorithm and finally it is (Imp-QMST).50% better than ICA algorithm.

Further, it can be seen that among all the swarm-based techniques, the ACO algorithm gives the highest throughput (203%, 177%, 203%, and 146% more as compared to PRIMS, PSO, FF and ICA algorithm for Imp-QMST) and it remains almost the same along with a varying number of nodes.

2) END-TO-END DELAY COMPARISON

The end-to-end delay between Q-MST and proposed Imp-QMST has been analyzed in Figure8. The calculation of end-to-end delay is performed using Eq. 7, which is mentioned in section III. The explicit values are shown in Table 7.

Figure 8 depicts that the end-to-end delay achieved by Imp-QMST is lower than the QMST approach. By analyzing the simulation results, shown in Table 7, it becomes clear that Imp-QMST is almost 9% better than the QMST approach. Regarding Imp-QMST, the ACO algorithm is almost 80% better than other algorithms viz. ACO, PSO, FF, and ICA algorithms.

Hence, it's clear by Fig. 8 that the ACO algorithm has a lower end-to-end delay as compared to other swarm intelligence approaches, depicted in the above-said figure.

3) POWER DISSIPATION COMPARISON

On the lines of the comparison performed in sub-section 1 and 2, a similar comparison related to power dissipation is shown here in Table 8 and Fig. 9.

It is clear from the tabular analysis that the average power dissipation while considering all the swarm intelligence algorithms, the Imp-QMST is 10% better than the QMST approach. Further, it is also apparent in Fig. 9 that power dissipated in the Imp-QMST is less than QMST. By the Imp-QMST approach with the ACO algorithm, it is 25% better than the PRIMS algorithm, 34% better than the PSO algorithm, 26% better than the FF algorithm, and 37% better than the ICA algorithm. Hence, it is concluded that among all the MST construction algorithms, the ACO algorithm has less power dissipation as compared to others and Imp-QMST performs better than the QMST approach.

The graphical representation of power dissipation is shown in Fig. 9.

4) DATA AGGREGATION USING VARIOUS SWARM INTELLIGENCE APPROACHES WHEN NUMBER OF NODES WERE KEPT FIXED (I.E. THE NUMBER OF NODES N = 35)

Here, the above-mentioned comparisons have been elaborated again by setting the count of nodes fixed to 35 and varying the call of data transmissions from 100 to 1000. The throughput, delay, and power dissipation comparisons are given in this section and subsections:

a: THROUGHPUT COMPARISON

A tabular comparison of throughput is shown in Table 9. Figure 10 narrates the throughput comparison between

TABLE 6. Throughput Comparison of QMST v/s Imp-QMST.

	PRIMS		ACO		PSO		FF		ICA	
NODES	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST
10	23.46	24.47	43.6	46.96	22.77	23.48	24.47	23.46	23.83	23.86
15	18.84	19.94	38.51	44.39	20.45	21.84	19.94	18.84	21.75	20.84
20	14.03	15.48	45.34	45.34	18.67	19.33	15.48	14.03	19.78	20.96
25	12.39	13.93	43.83	44.53	17.92	18.97	15.28	15.94	18.44	19.65
30	16.38	15.28	37.73	38.4	15.79	16.38	14.18	12.6	18.23	18.66
35	12.6	14.18	47.17	47.25	13.47	14.2	13.93	12.5	17.86	16.28
40	11.5	12.46	44.53	46.5	12.00	11.5	12.46	10.92	17.12	16.59
45	11.81	11.93	46.51	46.51	11.61	12.41	11.93	10.81	17.47	16.19
50	10.41	10.45	41.81	37.86	12.89	13.7	10.45	10.41	14.80	14.64
55	10.50	11.99	41.34	46.51	11.15	12.79	11.99	10.5	15.55	16.49
60	10.50	11.50	46.51	46.50	10.5	12.00	10.50	11.50	14.00	15.00

TABLE 7. End-to-End Delay Comparison of QMST v/s Imp-QMST.

	PRIMS		ACO		PSO		FF		ICA	
NODES	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST
10	1.50	1.51	0.82	0.63	1.517	1.47	1.38	1.51	1.52	1.47
15	2.13	2.01	0.94	0.78	2.13	2.01	2.41	2.07	2.13	2.01
20	2.96	2.87	0.69	0.69	3.05	2.87	2.97	2.93	3.01	2.91
25	3.03	2.99	0.78	0.73	3.45	3.36	2.88	2.88	3.10	3.02
30	4.32	3.59	0.98	0.9	4.37	3.54	4.57	3.84	3.69	3.23
35	4.6	4.3	0.65	0.65	4.26	3.64	5.24	4.75	4.20	3.96
40	4.85	4.15	0.82	0.68	4.95	4.49	5.58	4.95	4.82	3.59
45	4.68	4.18	0.68	0.68	5.2	4.95	5.00	4.51	5.23	4.53
50	4.95	4.6	0.91	0.94	5.5	5.0	5.64	4.60	5.95	4.85
55	5.48	5.06	0.90	0.67	5.8	5.6	5.7	5.10	6.08	5.5
60	5.59	5.50	0.98	0.90	6.2	5.9	6.0	5.2	7.1	6.95

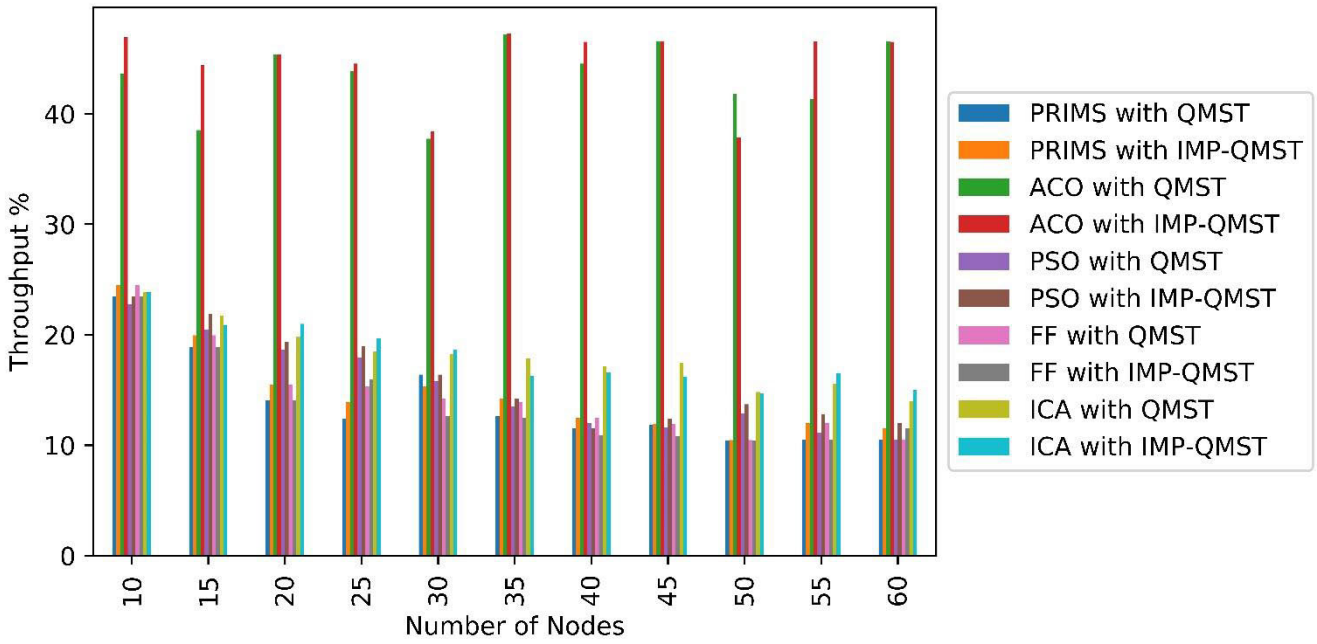

FIGURE 7. Throughput Comparison of QMST v/s Imp-QMST.

TABLE 8. Comparison of Energy Dissipation (in nJ) between QMST and Imp-QMST.

No. of Nodes	PRIMS		ACO		PSO		FF		ICA	
	QMST	Imp-QMST	QMST	Imp-QMST	QMST	Imp-QMST	QMST	Imp-QMST	QMST	Imp-QMST
10	1273521	1091707	422082	422082	1369757	1038710	1296064.07	1091707.37	1254395.68	981210.59
15	1374115	1165155	899151	436788	1374115	1165155	1409419.01	1176244.24	1374115.05	1165155.24
20	1539793	1353728	724105	684164	1604913	1331061	1444180.62	1263401.02	1308574.48	1356826.55
25	1302739	1318195	1083702	712403	1331061	1328611	1362940.97	1310744.21	1801851.8	1538917.86
30	1524726	1556288	1207680	884164	1595558	1288645	1540670.64	1403922.71	1670162.131	1546081.85
35	1601460	1387243	1105683	1090213	1528652	1372836	1482335.10	1402235.66	1648655.753	1671290.78
40	1787522	1423479	1143508	1243508	1842576	1457157	1744745.80	1605357.46	1941328.678	1849349.74
45	1836359	1510811	1300346	1200346	1950000	1560000	1880914.16	1670823.9	2189716.354	2077815.65
50	1989413	1793171	1592153	1307680	2115000	2112000	1968507.82	1703391.76	2338111.7	2187979.4
55	2068421	1855254	2151436	2063052	3040000	2700000	2048637.52	1911610.62	2605249.75	2609872.5
60	2270421	2093171	2581122	2320839	3562224	3474000	2344180.62	2063401.02	2708574.48	2656826.55

TABLE 9. Throughput Comparison when numbers of nodes are fixed.

No. of Transmissions	PRIMS		ACO		PSO		FF		ICA	
	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST
100	7.39	7.36	50.32	50.32	10.07	10	8.00	7.92	13.70	13.66
200	3.54	3.54	42.40	42.40	4.59	4.67	6.56	6.33	7.78	6.49
300	5.55	7.96	49.88	49.88	6.50	7.97	10.24	8.92	11.13	7.01
400	5.95	7.78	42.56	42.56	7.99	7.71	8.48	8.21	9.77	9.95
500	5.00	6.26	46.56	46.61	6.46	8.41	8.26	11.98	10.01	12.16
600	7.63	10.84	43.23	41.39	12.54	15.19	14.72	16.45	16.35	17.37
700	5.76	8.37	46.07	47.58	8.19	10.00	8.91	11.63	11.64	12.59
800	4.60	10.61	47.16	47.16	10.73	12.79	9.74	14.87	13.52	16.34
900	10.13	18.23	38.54	43.84	12.40	19.54	13.16	20.08	18.96	21.82
1000	11.72	17.92	45.68	47.16	13.35	19.28	15.85	19.89	18.12	21.63

TABLE 10. End-to-End delay Comparison when numbers of nodes are fixed.

No. of Transmissions	PRIMS		ACO		PSO		FF		ICA	
	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST
100	4.27	3.97	0.6	0.6	4.09	3.96	3.95	4.10	3.42	3.62
200	4.92	4.92	0.72	0.81	4.16	4.39	4.54	4.70	3.92	4.00
300	4.63	4.71	0.75	0.6	4.13	4.09	4.58	4.22	4.02	3.84
400	4.97	4.87	0.78	0.78	4.57	4.18	4.81	4.31	4.29	3.78
500	4.58	4.67	0.91	0.70	4.16	3.98	4.22	4.27	3.90	3.90
600	4.29	4.66	0.6	0.92	4.15	4.10	3.99	4.48	3.48	3.90
700	4.57	5.08	0.81	0.66	4.46	4.35	4.48	4.64	4.22	4.108
800	5.16	4.95	0.744	0.65	4.56	4.70	5.01	4.81	4.32	4.51
900	5.38	5.51	0.65	0.85	4.95	4.67	5.41	4.75	4.82	4.60
1000	4.99	5.52	1.08	0.75	4.86	4.85	4.94	5.30	4.54	4.90

QMST and Imp-QMST algorithms by using various artificial intelligence techniques as an MST generation algorithm (data aggregation).

The tabular analysis in (Table 9), shows that the cumulative throughput of proposed Imp-QMST applied to all the said algorithms is almost 11% better than the QMST approach. The individual analysis of each algorithm between Imp-QMST v/s QMST shows that the PRIMS algorithm with Imp-QMST is 48% better than with QMST. Similarly, ACO algorithms with Imp-QMST are 2% better than QMST, PSO algorithm with Imp-QMST is 24% better than QMST, FF algorithm with Imp-QMST is 21% efficient than QMST and finally, ICA algorithm with Imp-QMST is about 7% more, efficient than QMST approach. Further, regarding the Imp-QMST approach it has been found that the ACO algorithm performs 363% better than PRIMS, 295% better than the PSO algorithm, 264% better than the FF algorithm, and 230% better than ICA algorithm.

Hence, it is comprehensive that the throughput from the Imp-QMST procedure is higher than QMST while numbers

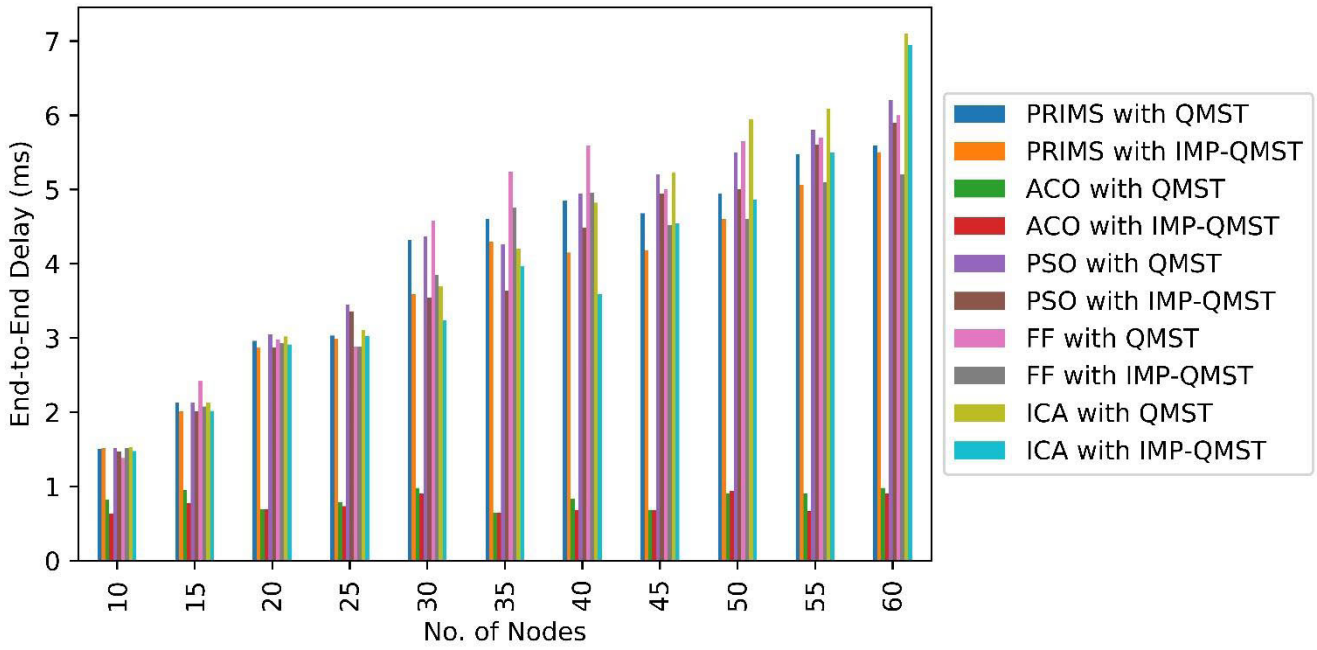

FIGURE 8. End-to-End Delay Comparison of QMST v/s Imp-QMST.

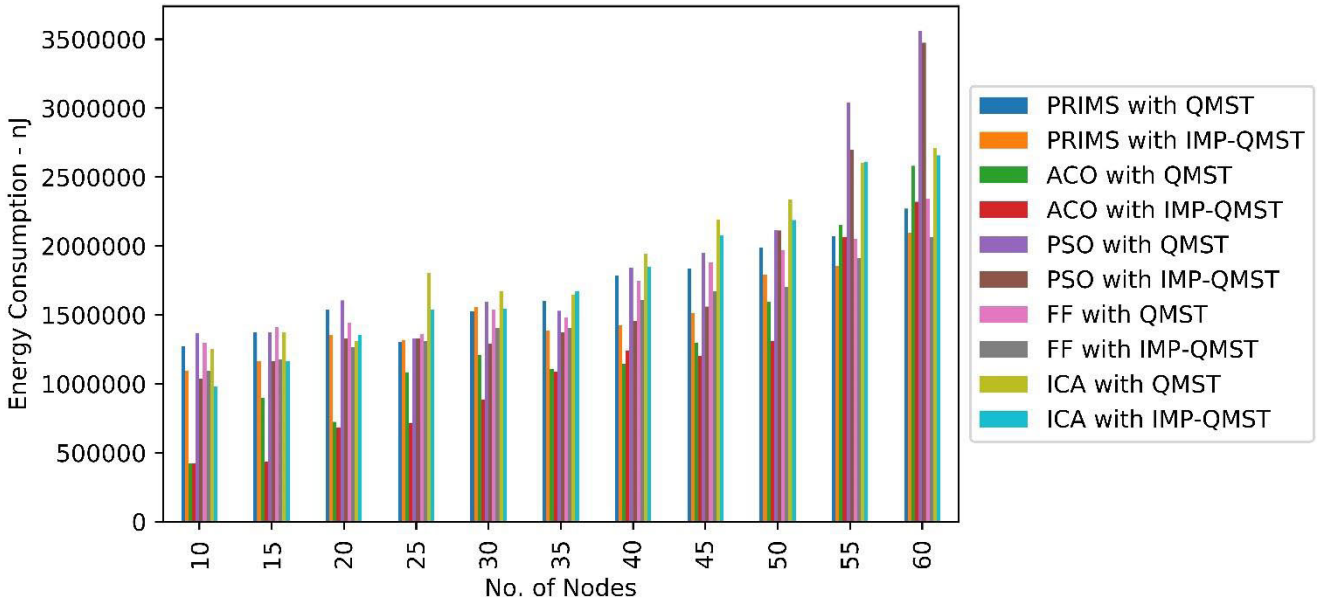

FIGURE 9. Comparison of Energy Dissipation between QMST and Imp-QMST.

of nodes are kept fixed and the ACO algorithm performs better than all the other said algorithms.

b: END-TO-END DELAY COMPARISON

The comparison regarding the delay between QMST and Imp-QMST by fixing the count of nodes is delineated in Table 10 and the graphical representation is shown in Fig. 11.

The analysis of Table 10 shows that the cumulative end-to-end delay of all the algorithms with Imp-QMST performs better. If we compare it individually on the ACO algorithm then it is found that the ACO algorithm with Imp-QMST performs almost 2% better than ACO with the QMST approach.

Further, it is clear from Fig. 11 that the Imp-QMST algorithm causes lesser delay than the QMST approach.

c: POWER DISSIPATION COMPARISON

Table 11 and Fig. 12 show that the energy consumed (nJ) by QMST is greater than Imp-QMST.

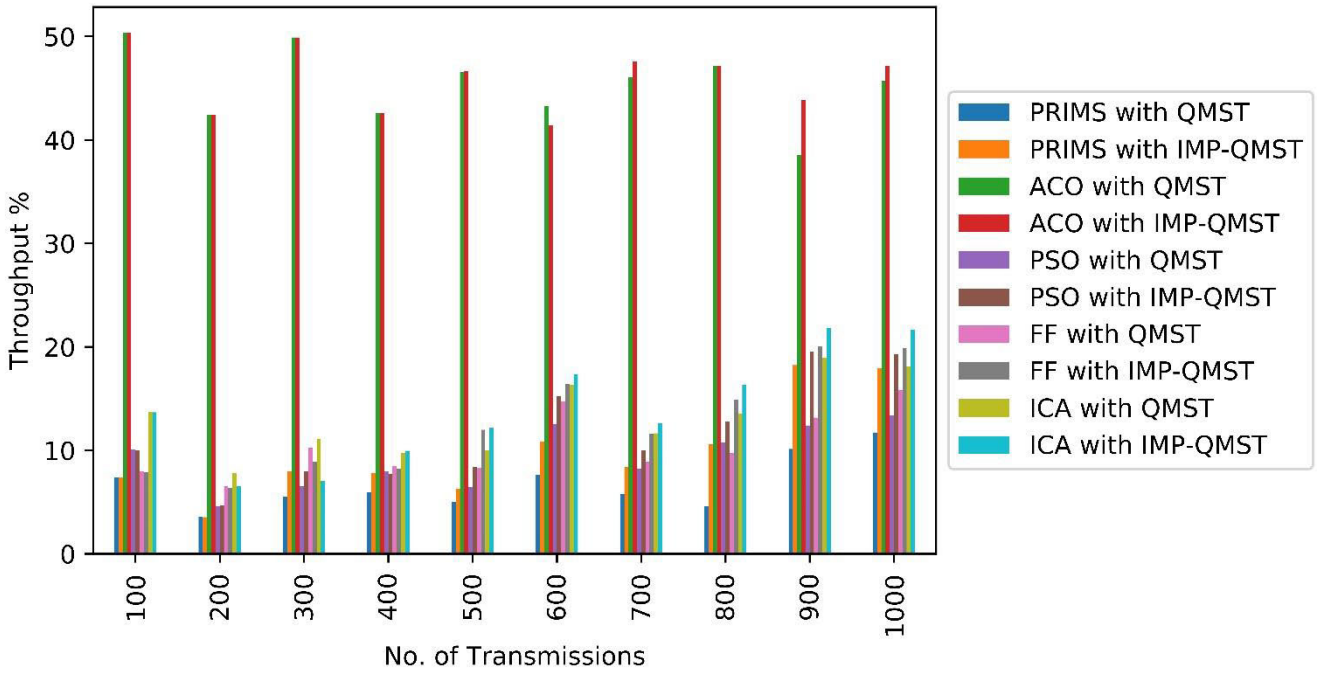

FIGURE 10. Throughput Comparison when numbers of nodes are fixed.

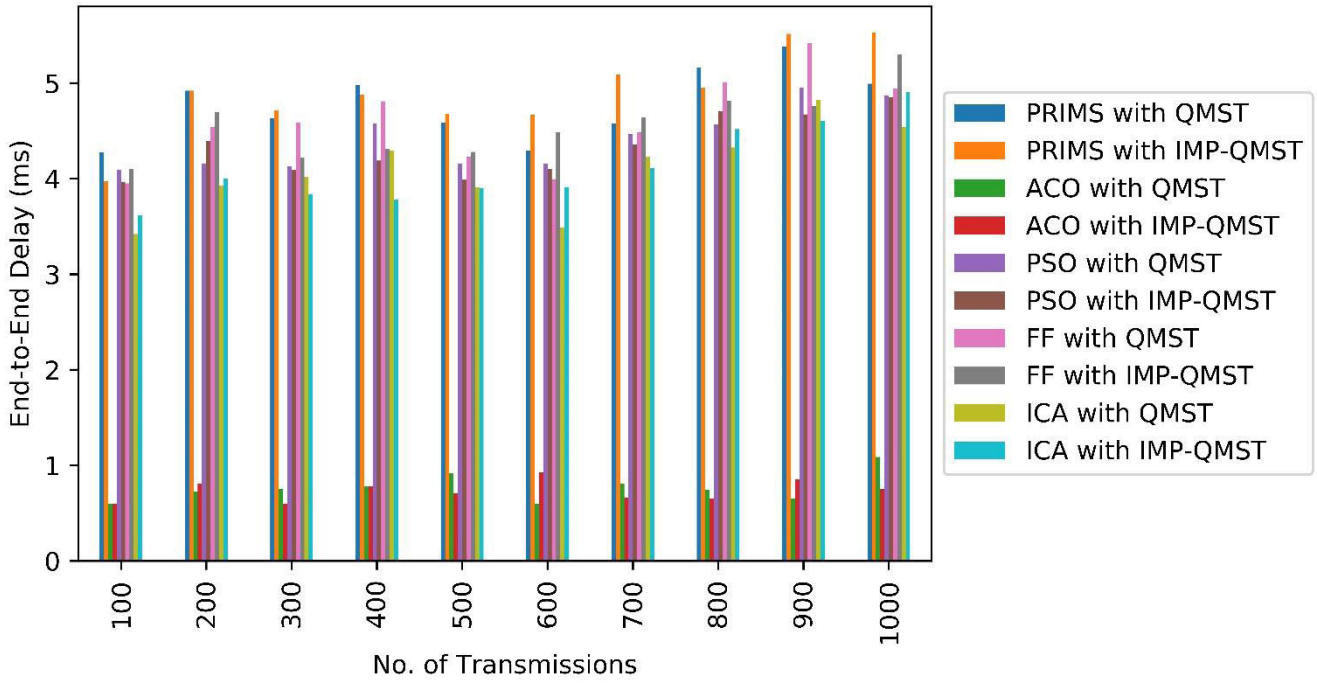

FIGURE 11. End-to-End delay Comparison when numbers of nodes are fixed.

Similar to previous analyses, here also the ACO algorithm performs better than to other algorithms i.e. in Imp-QMST, the ACO algorithm is about 32% better than PRIMS and PSO algorithm, 29% better than FF algorithm and 39% better than ICA algorithm.

B. HANDOFF VS IMP-HANDOFF ALGORITHM

Along the lines of the previous sub-section i.e. subsection A, the performance comparison between Handoff and proposed Imp-Handoff algorithm is measured on various artificial intelligence (AI) algorithms for three network parameters.

TABLE 11. Power Dissipation Comparison when numbers of nodes are fixed.

No. of transmissions	PRIMS		ACO		PSO		FF		ICA	
	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST	QMST	IMP-QMST
100	153867.05	180745.03	20000	20000	171311.5	168089.89	173372.8	171600.35	144329.4	140539.82
200	437079.73	422074.67	286315.4	286315.4	398747.6	386051.1	412543.3	384535.32	461834.6	430189.16
300	388550.33	440399.85	60000	150000	560765.6	575608.57	469689.4	388509.17	600622.9	575823.22
400	864349.93	704318	494148	494148	751872.7	768735.22	671318.6	651899.55	857649.3	804938.66
500	884220.78	996071.52	581592.2	435937.65	907332	907605.52	789219.7	878974.37	1096598	1156731.68
600	1068244.11	1074358.85	500808.8	809409.57	902814.1	976414.10	853064.3	925222.85	1122839	1203166.65
700	1253060.7	1307151.23	647371	950032	1186169	1164502.57	1123629	1203777.89	1467627	1399996.65
800	1707475.45	1705203.74	585176.7	985176.69	1492157	1638759.28	1378849	1514845.52	1805062	1779602.35
900	2009056.4	1923537.63	1007377	1505729.86	1730694	2011416.22	1562037	2060205.23	1997570	2107666.4
1000	2248990.9	2006235.61	1038741	1604970.4	1933507	2103332.37	1681567	2100325.3	2202850	2158064.74

TABLE 12. Throughput Comparison of Handoff vs. Imp-Handoff algorithm.

No of nodes	PRIMS		ACO		PSO		FF		ICA	
	Hand-Off	Imp-HandOff	Hand-Off	Imp-HandOff	Hand-Off	Imp-HandOff	Hand-Off	Imp-HandOff	Hand-Off	Imp-HandOff
10	21.66	21.66	42.74	42.99	20.28	20.83	20.81	20.81	20.28	20.83
15	20.02	19.94	40.19	41.19	20.1	19.94	19.32	20.30	20.10	19.94
20	13.36	13.74	44.18	45.18	18.26	17.43	12.78	13.16	19.42	18.34
25	12.84	11.55	44.53	45.53	16.13	15.4	16.14	15.71	17.27	17.46
30	14.3	14.68	38.3	38.3	14.97	13.02	12.37	13.23	16.14	15.36
35	13.07	14.93	45.63	45.97	13.54	13.82	12.58	11.73	13.37	15.23
40	11.61	12.85	46.5	46.5	11	11.52	10.76	10.42	12.76	12.12
45	11.73	12.5	46.51	46.51	13.61	13.65	8.83	10.64	11.83	11.00
50	9.93	10.37	38.43	39.43	11.87	12	9.65	9.46	11.25	12.39
55	9.44	10.72	43.77	43.77	10.64	11.64	9.00	9.60	11.00	12.50
60	10.02	10.5	46.3	46.3	11.5	11.95	10.4	11	12	14

1) THROUGHPUT COMPARISON

Table 12 and Fig. 13 enlist the throughput comparison between Handoff v/s IMP-Handoff algorithms on the data aggregation tree generated by various swarm intelligence techniques.

The tabular analysis makes it clear that the Imp-Handoff algorithm is better than handoff algorithm, tested on various AI algorithms i.e. Imp-Handoff algorithm is about 3% better than handoff algorithm for PRIMS algorithm, 1.5% better for ACO algorithm, 3% better for FF algorithm and 2% better for ICA algorithm.

To find a better-suited algorithm, the performance of all the said AI algorithms are compared and it is found that the performance of the ACO algorithm for generating MST along with Imp-Handoff algorithm for fault tolerance is best suited. The ACO algorithm with Imp-Handoff FT gives about 214% better throughput than PRIMS with the Imp-Handoff algorithm for FT. Similarly, ACO performs 198%, 229%, and 184% better than to PSO, FF, and ICA algorithms.

Hence, it is clear and it can also be seen from Fig. 13 that the throughput produced through the Imp-Handoff algorithm is way better than the Handoff algorithm.

2) END-TO-END DELAY COMPARISON

The tabular comparison between the handoff and the Imp-Handoff algorithm on various algorithms shown in Table 13.

Considering the end-to-end delay in Table 13, it is clear that on an average Imp-Handoff algorithm is about 9% better than the Handoff algorithm. If the PRIMS algorithm is considered then applying handoff and Imp-Handoff algorithms on it as FT techniques, the Imp-Handoff technique is around 7% better than the handoff algorithm. Similarly, Imp-handoff on ACO, FF, and ICA algorithm is found to be 11% better than handoff and on PSO it is found 7% better than the handoff algorithm.

Further, concerning Imp-Handoff, while comparing the performance of various algorithms, the ACO algorithm is found 80% efficient than PRIMS, PSO, FF, and ICA algorithm. Apart from the tabular comparison, the graphical representation of the end-to-end delay comparison of aforesaid is shown in Fig.14.

By observing Fig. 14, it is clear that the end-to-end delay using the Imp-Handoff algorithm is less than the Handoff algorithm. Further, it is also clear that as the number of nodes increases, the end-to-end delay also increases.

TABLE 13. End-to-End Delay Comparison using Handoff and Imp-Handoff algorithm.

No. of nodes	PRIMS		ACO		PSO		FF		ICA	
	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff
10	1.53	1.53	0.78	0.78	1.62	1.57	1.63	1.62	1.62	1.57
15	1.84	1.86	0.84	0.84	1.86	1.84	1.96	1.86	1.84	1.86
20	2.71	2.64	0.72	0.72	2.8	2.72	2.77	2.7	2.81	2.89
25	3.28	3.05	0.73	0.73	3.21	3.15	3.36	2.89	2.88	2.72
30	3.3	3.29	0.92	0.92	3.38	3.32	3.33	3.47	3.07	2.98
35	4.07	3.73	0.7	0.7	3.43	2.98	3.93	4.03	3.43	3.48
40	3.98	3.69	0.68	0.68	4.14	4.26	4.51	4.35	3.725	3.70
45	3.87	3.73	0.68	0.68	4.6	4.3	4.30	4.24	3.84	3.79
50	4.18	3.86	0.78	0.9	5.6	5.2	4.40	4.3	4.97	4.14
55	4.55	4.05	0.75	0.75	5.75	5.4	4.52	4.4	5.15	4.94
60	4.8	4.4	0.95	0.95	6	5.98	5	4.6	5.95	5.5

TABLE 14. Power Dissipation Comparison between Handoff and Imp-Handoff algorithm.

No. of nodes	PRIMS		ACO		PSO		FF		ICA	
	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff	Handoff	Imp-Handoff
10	2492021	2391623	462050	443276	2530670	2455710	2428408.52	2463432.7	2544357.97	2473153.53
15	2754155	2441490	827020	875379	2631118	2588016	2885225.06	2473154.9	2754155.04	2441489.77
20	2840292	2541119	1092722	909764	2754155	2441490	2647617.78	2548008.97	2652437.26	2410124
25	2917492	2687878	1092722	959848	2713952	3019215	2688757.35	2559292.12	2851537.79	3232695.04
30	3382745	2703484	1182434	1242824	3423210	3275191	3168943.65	2808571.71	3291597.81	3002076.96
35	3405562	2932499	1089742	1209764	3489407	3084088	3280859.07	2782611.80	3424195.39	3054103.17
40	3575977	3398145	1343508	1243508	3471749	3394396	3645523.53	3515301.73	3633655.55	3372037.16
45	3864311	3599702	1454018	1388359	3590210	3290191	3776095.12	3287019.33	3861026.43	3873252.92
50	4243743	3758101	1971775	1939955	3790210	3332159	3972132.11	3445324.02	3916967.31	3645621.72
55	4790779	4439574	2274418	2153286	3824691	3866317	4076372.87	3601801.67	4157817.42	4001112.82
60	5000000	4440850	2535491	2723233	4090210	3890191	4547617.78	4148008.97	4252437.26	4202076.96

3) POWER DISSIPATION COMPARISON

The comparison of power dissipated in the fault tolerance process by using both the Handoff and the proposed Imp-Handoff algorithms on various AI algorithms is visualized in Table 14. Further, graphical analysis of shown comparison is shown in Fig. 15.

Table 14 shows that cumulatively considering all the AI algorithms shown in this table, the proposed Imp-Handoff algorithm is 7% more efficient in energy consumption as compared to the existing Handoff algorithm. If we compare the Handoff and Imp-Handoff algorithm on an individual algorithm, then it is found that, for the PRIMS algorithm, the Imp-Handoff algorithm is 10% better than the Handoff algorithm. Similarly, for ACO, PSO, FF, and ICA algorithm, the Imp-Handoff is 2%, 5%, 9%, and 4% better than the Handoff mechanism.

Further, to find out the best-suited algorithm to construct MST respective to the Imp-Handoff mechanism, it is found that the ACO algorithm (Table 14) is about 56% efficient than PRIMS, PSO, FF, and ICA algorithm.

Figure 15 also makes it clear that during transmission, the power dissipated by the traditional Handoff algorithm is higher than the Imp-Handoff algorithm. Power consumption is a crucial parameter in any network as it directly affects network lifetime.

4) DATA AGGREGATION USING VARIOUS SWARM INTELLIGENCE APPROACHES WHEN NUMBER OF NODES WERE KEPT FIXED (I.E. THE NUMBER OF NODES N = 35)

Along the lines of section 4 of subsection A of result analysis section i.e. section VII-A-4, an analysis between Handoff and Imp-Handoff is demonstrated here by adjusting

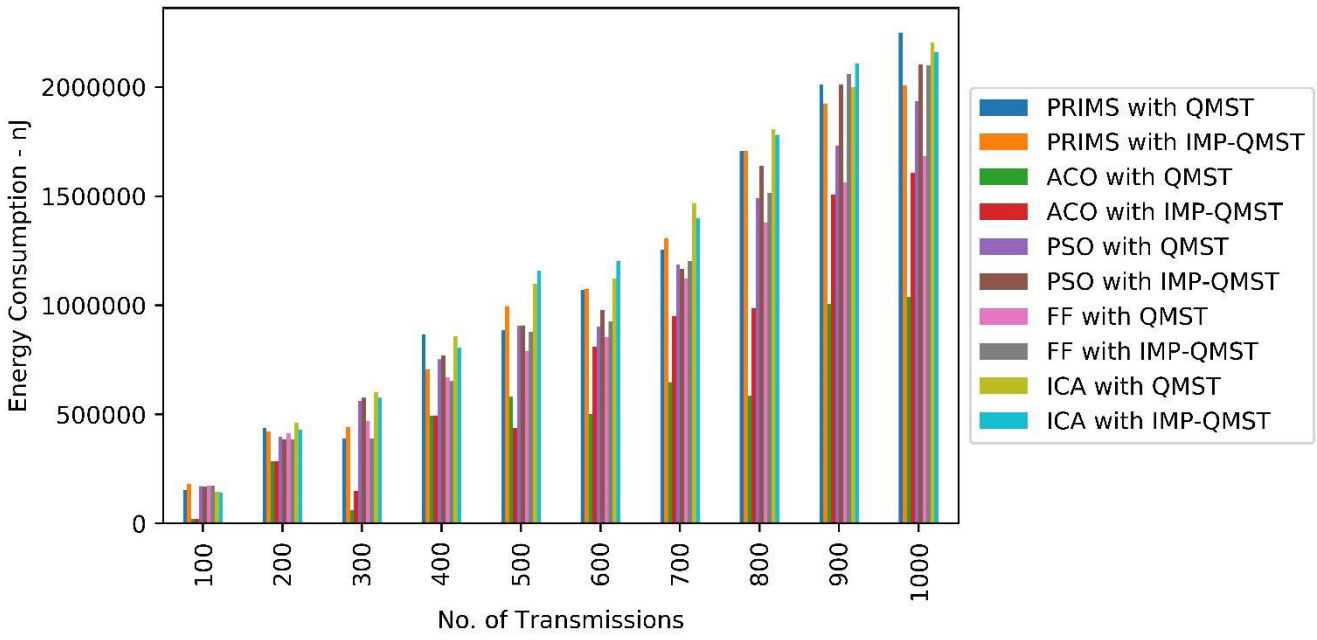

FIGURE 12. Power Dissipation Comparison when numbers of nodes are fixed.

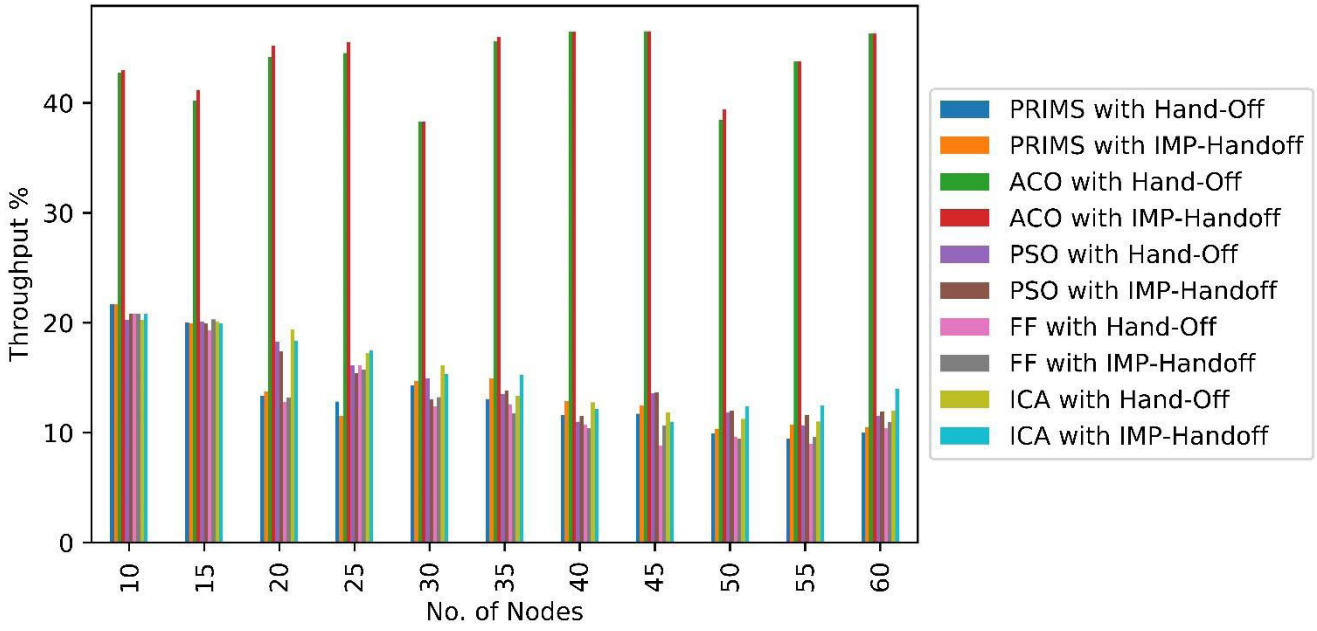

FIGURE 13. Throughput Comparison of Handoff vs. Imp-Handoff algorithm.

the count of nodes to 35 and varying the number of transmissions.

a: THROUGHPUT COMPARISON

Table 15 and Fig. 16 in this section enlist the throughput comparison between the Handoff and Imp-Handoff algorithm by keeping the fixed number of nodes and increasing the number of transmissions.

The tabular comparison makes it clear that the cumulative throughput of the Imp-Handoff mechanism is about 3-4% more than the Handoff mechanism. The individual comparison of various algorithms by the Imp-Handoff mechanism shows that the Imp-Handoff applied to the PRIMS algorithm is about 28% efficient than the Handoff algorithm. Similarly, the Imp-Handoff applied on ACO, PSO, FF, and PSO algorithms are about 1%, 19%, 14%, and 9% efficient than the Handoff algorithm.

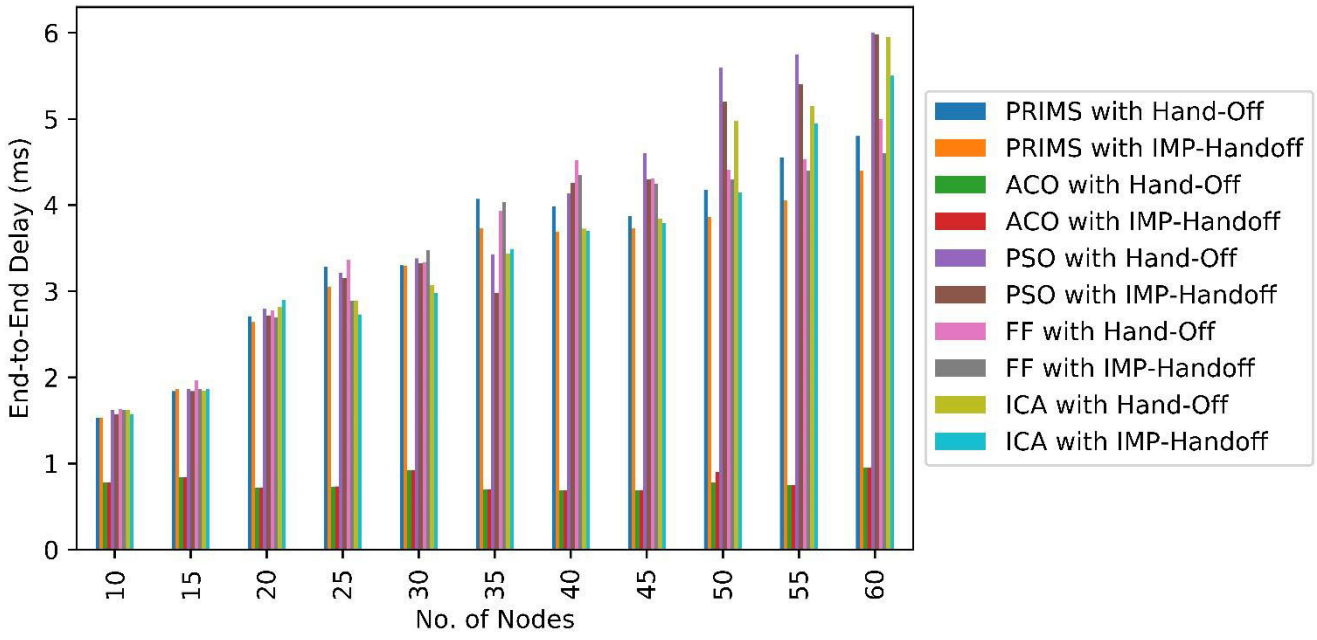

FIGURE 14. End-to-End Delay Comparison using Handoff and Imp-Handoff algorithm.

TABLE 15. Throughput Comparison between Handoff and Imp-Handoff Algorithm when numbers of nodes are fixed.

NO. OF TRANSMISSIONS	PRIMS		ACO		PSO		FF		ICA	
	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF
100	4.26	7.30	50.32	50.32	7.11	9.64	9.96	8.07	13.69	10.02
200	5.04	5.04	41.55	41.55	7.91	4.45	8.07	8.62	8.86	10.84
300	5.40	5.17	49.88	49.88	9.25	6.12	7.41	9.50	9.57	11.92
400	7.04	7.46	42.56	42.56	8.44	9.27	9.11	9.54	11.32	11.84
500	7.36	7.64	47.00	47.20	9.29	8.22	10.07	10.50	11.95	12.58
600	13.08	11.50	41.99	43.36	12.62	12.54	13.89	13.35	16.01	15.41
700	7.77	12.64	47.58	47.58	14.90	9.22	12.04	15.48	15.20	17.11
800	7.39	14.05	45.44	45.44	15.01	9.30	11.47	17.48	12.51	18.75
900	10.92	14.90	42.07	42.07	15.01	12.50	14.68	16.17	16.10	17.18
1000	9.94	14.50	45.04	45.041	17.12	12.98	14.55	17.89	16.33	18.67

Further, it can also be concluded from Table 15 that for Imp-Handoff, the ACO algorithm is around 355% efficient than the PRIMS algorithm, 350% efficient than PSO, 261% than FF, and 215% efficient than ICA algorithm.

Hence, it can be concluded that by keeping the fixed number of nodes and changing the transmissions in ascending order, the throughput by the Imp-Handoff algorithm becomes better than the Handoff algorithm.

b: END-TO-END DELAY COMPARISON

The end-to-end delay comparison of the Handoff algorithm and the Imp-Handoff algorithm with said data aggregation algorithms by setting the count of nodes to 35, has been given in Table 16. The pictorial analysis of the end-to-end delay comparison of Handoff and Imp-handoff algorithms is shown in Fig. 17.

The analysis of Table 16 shows that the Imp-Handoff algorithm performs better and, on average, it is about 2% better as compared to the Handoff algorithm. By comparing individual algorithms with Imp-Handoff and Handoff, it is found that Imp-Handoff applied on the PRIMS algorithm gives about 2% better performance, ICA gives 8% better performance and rest of the algorithms mentioned in Table 16 give a slightly better or equal performance as compared to

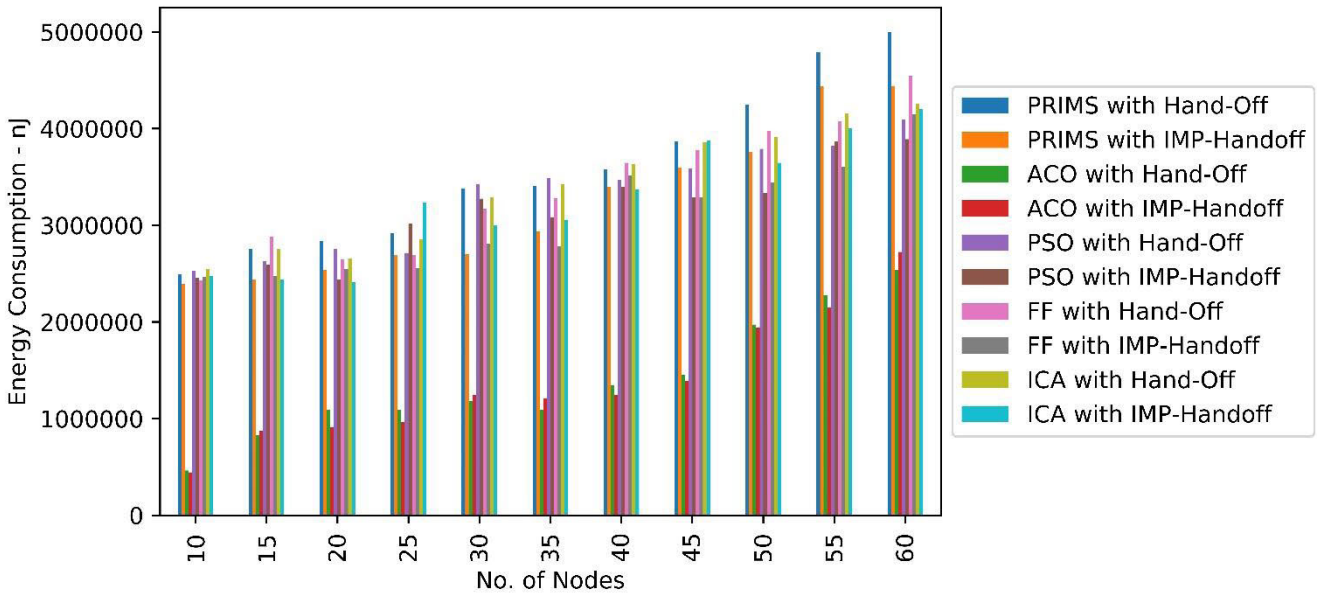

FIGURE 15. Power Dissipation Comparison between Handoff and Imp-Handoff algorithm.

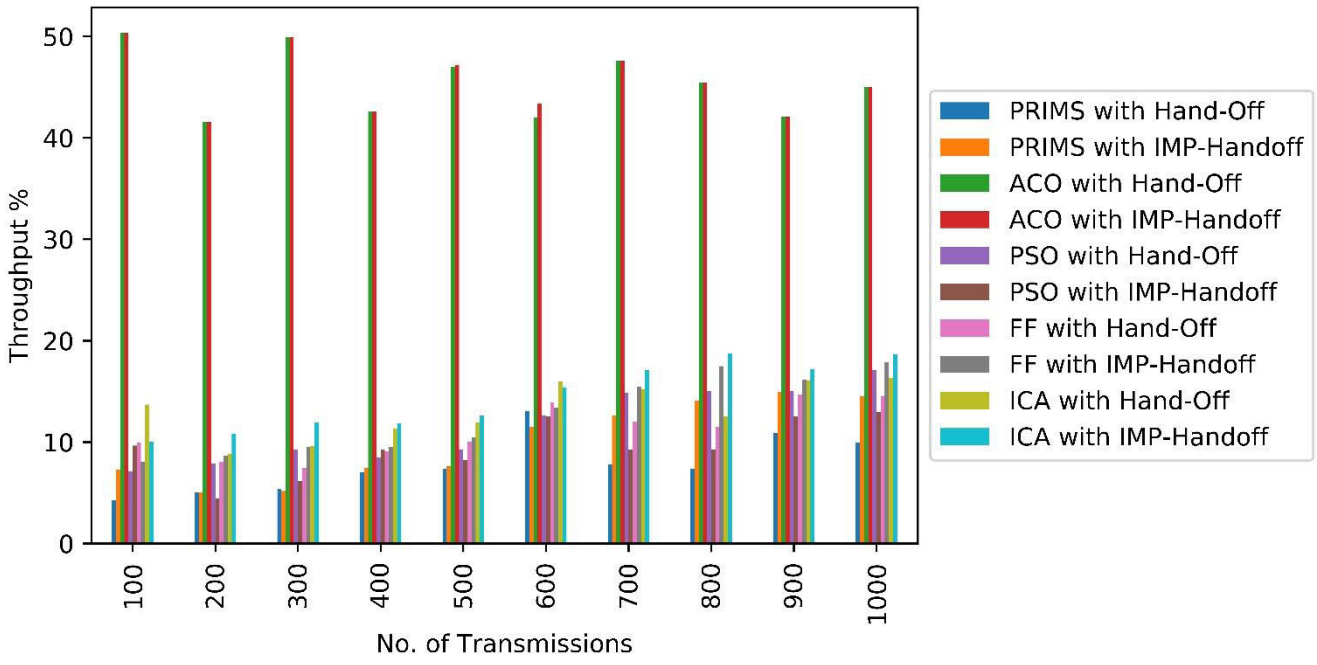

FIGURE 16. Throughput Comparison between Handoff and Imp-Handoff Algorithm.

Handoff approach. Further, by analyzing Table 16, it becomes clear that by Imp-Handoff, the ACO algorithm is an average 82% efficient as compared to PRIMS, PSO, FF, and ICA algorithms.

Hence, Fig. 17 concludes that the end-to-end delay of the Imp-Handoff algorithm is way lesser than the Handoff algorithm and stays almost the same for all the number of transmissions, for all MST generation algorithms.

c: POWER DISSIPATION COMPARISON

The power dissipation comparison is also specified in Table 17. Table 17 depicts that, on an average, Imp-Handoff algorithm is 8% efficient in saving energy as compared to the Handoff algorithm. The Imp-Handoff on PRIMS algorithm saves 5% energy as compared to the Handoff algorithm. Similarly, the Imp-Handoff algorithm, as a fault tolerance algorithm applied on ACO, PSO, FF, and ICA;

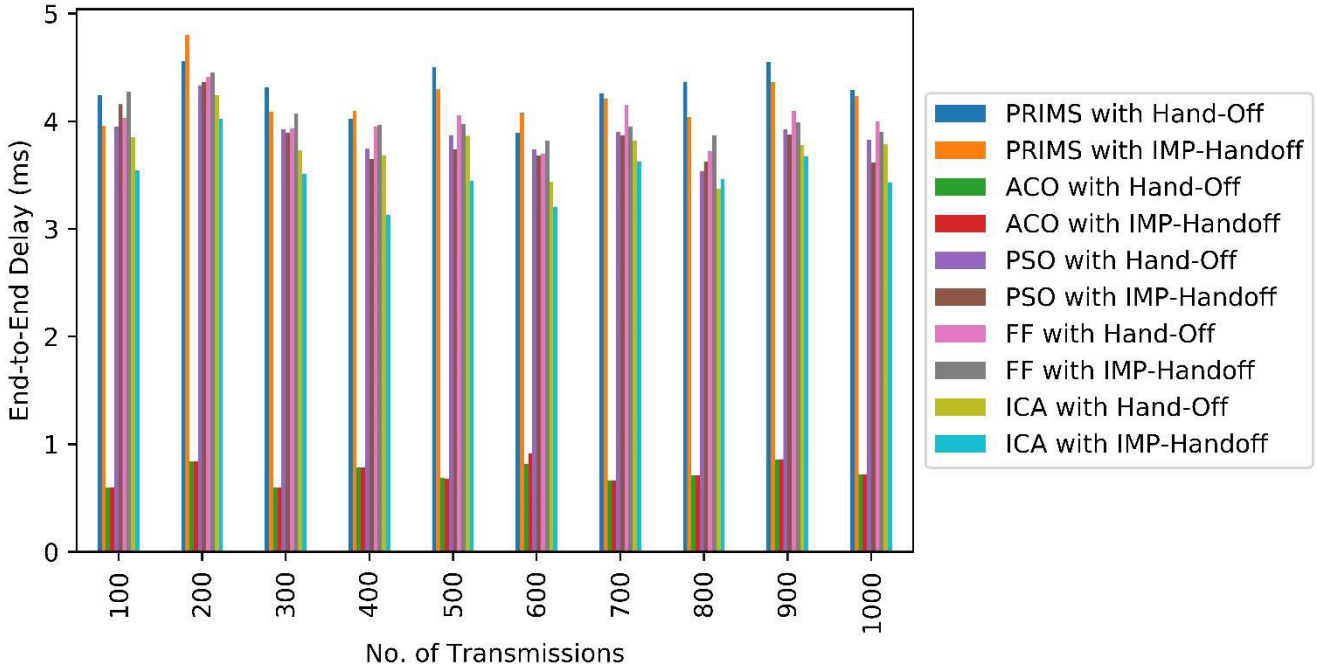

FIGURE 17. End-to-End Delay Comparison of the Handoff v/s Imp-Handoff Algorithm.

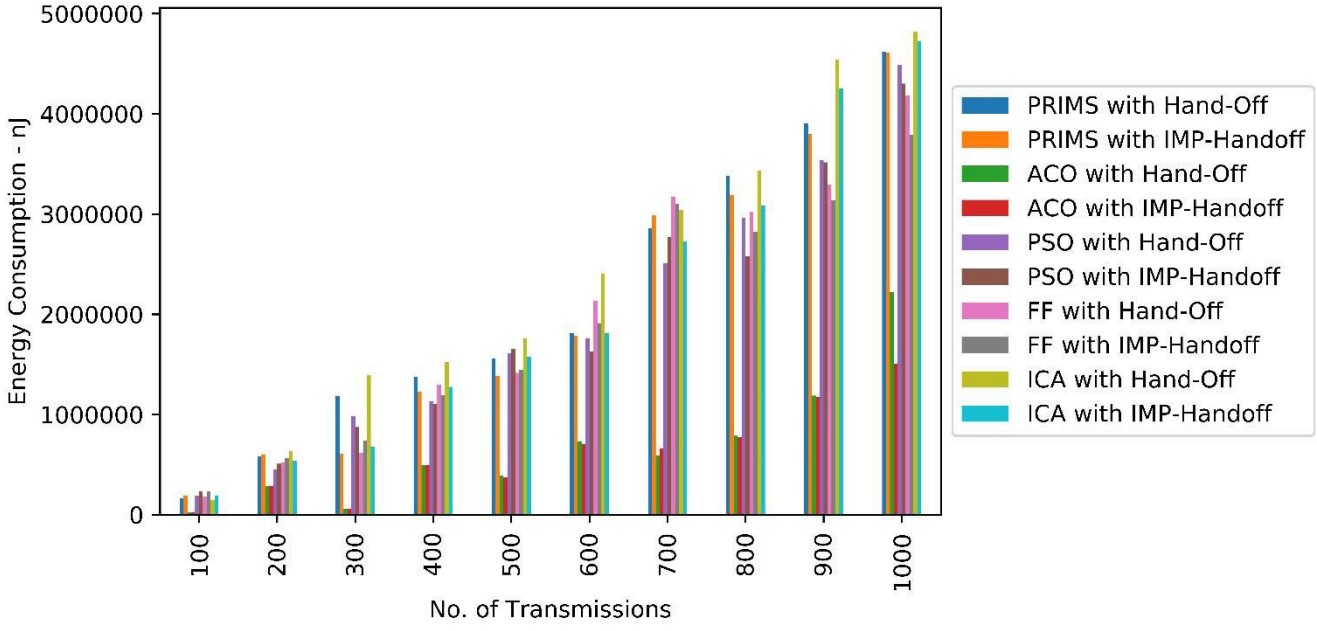

FIGURE 18. Power Dissipation Comparison between the Handoff and Imp-Handoff algorithm.

saves about 11%, 2%, 5%, and 12% less energy as compared to the Handoff mechanism. Similar to previous result analysis sections, Fig. 18 depicts the graphical analysis of power dissipation between handoff and Imp-Handoff techniques on MST constructed by various artificial intelligence techniques.

Further, the Imp-Handoff mechanism applied to the ACO algorithm is around 68% energy efficient as compared to other algorithms viz. PRIMS, PSO, FF, and ICA algorithm.

Hence, by viewing Fig. 18 and Table 17, it is concluded that the power dissipation by the Imp-Handoff mechanism is low as compared to the existing handoff mechanism.

TABLE 16. End-to-End Delay Comparison between Handoff and Imp-Handoff Algorithm when numbers of nodes are fixed.

NO. OF TRANSMISSIONS	PRIMS		ACO		PSO		FF		ICA	
	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF
100	4.24	3.96	0.6	0.6	3.95	4.15	4.03	4.27	3.85	3.54
200	4.55	4.8	0.84	0.84	4.33	4.36	4.41	4.45	4.24	4.02
300	4.31	4.09	0.6	0.6	3.92	3.89	3.93	4.07	3.73	3.51
400	4.02	4.094	0.78	0.78	3.74	3.64	3.95	3.96	3.68	3.13
500	4.5	4.3	0.69	0.68	3.86	3.73	4.05	3.97	3.86	3.44
600	3.89	4.07	0.81	0.91	3.73	3.67	3.70	3.82	3.43	3.20
700	4.25	4.21	0.66	0.66	3.86	3.9	4.15	3.95	3.82	3.62
800	4.36	4.03	0.70	0.70	3.53	3.62	3.72	3.86	3.37	3.45
900	4.54	4.36	0.85	0.85	3.92	3.87	4.09	3.99	3.77	3.67
1000	4.28	4.23	0.72	0.72	3.82	3.61	3.99	3.90	3.78	3.42

TABLE 17. Power Dissipation Comparison between the Handoff and Imp-Handoff algorithm when numbers of nodes are fixed.

NO. OF TRANSMISSIONS	PRIMS		ACO		PSO		FF		ICA	
	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF	HANDOFF	IMP-HANDOFF
100	161155.29	189969.04	20000	20000	186562	235147.94	179380.2	235920.4	148563.9	188513.96
200	584841.79	596943.35	283936.8	283936.8	451065.6	506906.72	523593.2	566790.28	635470.8	534311.24
300	1180049.92	608003.31	60000	60000	981046.4	873574.2	616170.9	740031.01	1389977	679489.84
400	1375648.66	1227094.47	494148	494148	1128103	1106779.65	1298195	1193615.98	1526341	1274368.35
500	1561695.59	1382153.87	385125.8	370390.78	1611960	1651007.41	1413952	1444483.64	1754518	1574527.26
600	1814506.17	1785438.74	732649.9	701963.78	1757781	1628506.64	2133738	1911637.71	2402584	1811521.32
700	2854913.63	2984716.38	593371.2	664051.62	2508204	2768046.51	3172278	3103220.43	3040416	2726487.98
800	3381550.51	3188843.83	790120.9	772572.12	2958723	2577411.22	3022665	2818754.81	3429592	3087534.51
900	3908400.32	3801764.48	1187423	1170712.75	3535823	3515822.51	3294745	3139388.47	4540862	4255604.32
1000	4623295.89	4608897.77	2219400	1504597.15	4490298	4300973.96	4182789	3787592.56	4818306	4725489.93

VIII. CONCLUSION AND FUTURE SCOPE

The result analysis of section VII makes it clear that the proposed NLFFT model for fault tolerance (shown in section IV) is successful when any node or link failure occurs in the sensor networks. The supportive Imp-QMST and Imp-Handoff algorithms are found to be way better than QMST and Hand-off algorithm in terms of throughput, end-to-end delay, and power dissipation. In the mentioned result analysis section, four swarm intelligence techniques (ACO, PSO, FF, and ICA) and one traditional PRIMS algorithm have been applied to generate MST to support data aggregation, and then fault tolerance mechanism is imposed according to the proposed NLF fault-tolerance model.

Simulation results depict that as compared to basic techniques, i.e. Q-MST and Handoff algorithm, the proposed NLFFT model improvises the performance of WSN around 7%. The comparison of the individual algorithm shows that the Imp-QMST gives about 6% improved throughput, 5% lesser end-to-end delay, and 6% less power consumption as compared to the QMST algorithm. Similarly, Imp-Handoff improves about 4% throughput, 6% lesser end-to-end delay, and 7% less power consumption as compared to the Handoff algorithm. Hence, the proposed NLFFT model proves the improved performance in terms of throughput, end-to-end delay, and power consumption.

Though the complexity of the proposed algorithms in the NLFFT model is increased by involving more than one parameter, i.e. battery power, cost of edges but the competent results go on to prove the efficacy of the proposed NLFFT model and shows that it can be applied to handle node or link failures.

Further, as the proposed NLFFT fault tolerance model is applied to the statically deployed sensor nodes only, it can be extended and applied to the dynamically deployed

sensor networks also. This work can also be extended in Internet-based applications by adding some security protocols and regulations.

REFERENCES

[1] S. Kim, J. Ko, J. Yoon, and H. Lee, "Energy-efficient and fault-tolerant positioning of multiple base stations," in *Proc. Int. Conf. Inf. Netw.* Berlin, Germany: Springer, 2007, pp. 584–593.

[2] W. M. Elsayed, S. F. Sabbeh, and A. M. Riad, "A distributed fault tolerance mechanism for self-maintenance of clusters in wireless sensor networks," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 6891–6907, Dec. 2018.

[3] V. K. Menaria, S. C. Jain, and A. Nagaraju, "A fault tolerance based route optimisation and data aggregation using artificial intelligence to enhance performance in wireless sensor networks," *Int. J. Wireless Mobile Comput.*, vol. 14, no. 2, pp. 123–137, 2018.

[4] S. Sundar and A. Singh, "A swarm intelligence approach to the quadratic minimum spanning tree problem," *Inf. Sci.*, vol. 180, no. 17, pp. 3182–3191, Sep. 2010.

[5] P. Nehra and A. Nagaraju, "Fault tolerance using quadratic-minimum spanning tree (Q-MST) with secure data aggregation in wireless sensor networks," in *Proc. 14th IEEE India Council Int. Conf. (INDICON)*, Dec. 2017, pp. 1–6.

[6] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 738–754, Jun. 2006.

[7] M.-H. Lee and Y.-H. Choi, "Fault detection of wireless sensor networks," *Comput. Commun.*, vol. 31, no. 14, pp. 3469–3475, Sep. 2008.

[8] W. Qu, M. Kitsuregawa, H. Shen, and Z. Shan, "A novel fault-tolerant execution model by using of mobile agents," *J. Netw. Comput. Appl.*, vol. 32, no. 2, pp. 423–432, Mar. 2009.

[9] L. Guo, L. Zhang, Y. Peng, J. Wu, X. Zhang, W. Hou, and J. Zhao, "Multi-path routing in spatial wireless ad hoc networks," *Comput. Electr. Eng.*, vol. 38, no. 3, pp. 473–491, May 2012.

[10] D. D. Geeta, N. Nalini, and R. C. Biradar, "Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach," *J. Netw. Comput. Appl.*, vol. 36, no. 4, pp. 1174–1185, Jul. 2013.

[11] S. Abba and J.-A. Lee, "An autonomous self-aware and adaptive fault tolerant routing technique for wireless sensor networks," *Sensors*, vol. 15, no. 8, pp. 20316–20354, Aug. 2015.

[12] Y. Zhou, X. Wang, T. Wang, B. Liu, and W. Sun, "Fault-tolerant multi-path routing protocol for WSN based on HEED," *Int. J. Sensor Netw.*, vol. 20, no. 1, pp. 37–45, 2016.

[13] A. V. Sutagundar, V. S. Bennur, A. M. Anusha, and K. N. Bhanu, "Agent based fault tolerance in wireless sensor networks," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, vol. 1, Aug. 2016, pp. 1–6.

[14] N. Tien, S. Kim, J. Rhee, and S. Park, "A novel dual separate paths (DSP) algorithm providing fault-tolerant communication for wireless sensor networks," *Sensors*, vol. 17, no. 8, p. 1699, Jul. 2017.

[15] S. Mitra and A. Das, "Distributed fault tolerant architecture for wireless sensor network," *Informatica*, vol. 41, no. 1, pp. 1–12, 2017.

[16] G. Ma, Y. Yang, X. Qiu, Z. Gao, and H. Li, "Fault-tolerant topology control for heterogeneous wireless sensor networks using multi-routing tree," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 620–623.

[17] B. A. Begum and S. V. Nandury, "Component-based self-healing algorithm with dynamic range allocation for fault-tolerance in WSN," in *Proc. 7th Int. Conf. Comput. Commun. Technol. (ICCCT)*, 2017, pp. 58–65.

[18] D. Kim, Y. Kim, D. Li, and J. Seo, "A new maximum fault-tolerance barrier-coverage problem in hybrid sensor network and its polynomial time exact algorithm," *Ad Hoc Netw.*, vol. 63, pp. 14–19, Aug. 2017.

[19] S. Henna, "Energy efficient fault tolerant coverage in wireless sensor networks," *J. Sensors*, vol. 2017, pp. 1–11, Apr. 2017.

[20] N. Moussa and A. E. Elalaoui, "Fault tolerant routing protocols in wireless sensor networks: A decision support tool," *Int. J. Wireless Mobile Comput.*, vol. 10, no. 4, pp. 361–370, 2016.

[21] S. J. Jassbi and E. Moridi, "Fault tolerance and energy efficient clustering algorithm in wireless sensor networks: FTEC," *Wireless Pers. Commun.*, vol. 107, no. 1, pp. 373–391, Jul. 2019.

[22] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[23] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Comput. Netw.*, vol. 42, no. 6, pp. 697–716, Aug. 2003.

[24] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, p. 10.

[25] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[26] F. Neumann and C. Witt, "Runtime analysis of a simple ant colony optimization algorithm," in *Proc. Int. Symp. Algorithms Comput.* Berlin, Germany: Springer, 2006, pp. 618–627.

[27] F. Neumann and C. Witt, "Ant colony optimization and the minimum spanning tree problem," *Theor. Comput. Sci.*, vol. 411, no. 25, pp. 2406–2413, May 2010.

[28] C. Lin, G. Wu, F. Xia, M. Li, L. Yao, and Z. Pei, "Energy efficient ant colony algorithms for data aggregation in wireless sensor networks," *J. Comput. Syst. Sci.*, vol. 78, no. 6, pp. 1686–1702, Nov. 2012.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[30] E. F. G. Goldbarg, G. R. de Souza, and M. C. Goldbarg, "Particle swarm optimization for the bi-objective degree constrained minimum spanning tree," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 420–427.

[31] P. Bhambu, S. Kumar, and K. Sharma, "Self balanced particle swarm optimization," *Int. J. Syst. Assurance Eng. Manage.*, vol. 9, no. 4, pp. 774–783, 2018.

[32] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.

[33] X.-S. Yang and X. He, "Firefly algorithm: Recent advances and applications," *Int. J. Swarm Intell.*, vol. 1, no. 1, pp. 36–50, 2013.

[34] S. Kumar and R. Kumari, "Artificial bee colony, firefly swarm optimization, and bat algorithms," in *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*. Boca Raton, FL, USA: CRC Press, 2018, pp. 145–182.

[35] G. Yogarajan and T. Revathi, "Nature inspired discrete firefly algorithm for optimal mobile data gathering in wireless sensor networks," *Wireless Netw.*, vol. 24, no. 8, pp. 2993–3007, Nov. 2018.

[36] I. Mosavvar and A. Ghaffari, "Data aggregation in wireless sensor networks using firefly algorithm," *Wireless Pers. Commun.*, vol. 104, no. 1, pp. 307–324, Jan. 2019.

[37] S. Hosseini and A. Al Khaled, "A survey on the imperialist competitive algorithm Metaheuristic: Implementation in engineering domain and directions for future research," *Appl. Soft Comput.*, vol. 24, pp. 1078–1094, Nov. 2014.

[38] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 4661–4667.

[39] S. Kumar, S. Jain, and H. Sharma, "Genetic algorithms," in *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*. Boca Raton, FL, USA: CRC Press, 2018, pp. 27–52.

[40] M. H. Sayadnavard, A. T. Haghighat, and M. Abdechiri, "Wireless sensor network localization using imperialist competitive algorithm," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, vol. 9, Jul. 2010, pp. 818–822.

[41] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.

[42] A. Assad and W. Xu, "The quadratic minimum spanning tree problem," *Nav. Res. Logistics*, vol. 39, no. 3, pp. 399–417, Apr. 1992.

[43] A. Nayyar and R. Singh, "A comprehensive review of simulation tools for wireless sensor networks (WSNs)," *J. Wireless Netw. Commun.*, vol. 5, no. 1, pp. 19–47, 2015.

[44] A. Kumar and A. Nayyar, "Energy efficient routing protocols for wireless sensor networks (WSNs) based on clustering," *Int. J. Sci. Eng. Res.*, vol. 5, no. 6, pp. 440–448, 2014.

[45] A. Nayyar and R. Singh, "Ant colony optimization (ACO) based routing protocols for wireless sensor networks (WSN): A survey," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 2, pp. 148–155, 2017.

[46] A. Nayyar and R. Singh, "Ant colony optimization—Computational swarm intelligence technique," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Development (INDIACom)*, Mar. 2016, pp. 1493–1499.

[47] A. Nayyar, A. D. N. Le, and N. G. Nguyen, *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*. Boca Raton, FL, USA: CRC Press, 2018.

[48] A. Nayyar and N. G. Nguyen, "Introduction to swarm intelligence," in *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*. Boca Raton, FL, USA: CRC Press, 2018, pp. 53–78.
[49] A. Nayyar, S. Garg, D. Gupta, and A. Khanna, "Evolutionary computation: Theory and algorithms," in *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*. London, U.K.: Chapman & Hall/CRC, 2018, pp. 1–6.

VINOD KUMAR MENARIA received the M.Tech. degree in computer science and engineering. He is currently pursuing the Ph.D. degree in CSE with Rajasthan Technical University, Kota, India. His research interests include wireless sensor networks, the Internet of Things, cloud computing, high-performance computing, and swarm intelligence.

S. C. JAIN received the master's degree in computer science and technology from IIT Roorkee and the Ph.D. degree in VLSI design from IIT Delhi. He has served the Defense Research and Development Organization, Bengaluru, India. He is currently working as a Professor with the Computer Science and Engineering Department, Rajasthan Technical University, Kota, India. His research interests include VLSI design, real time embedded system, reversible computing, wireless sensor networks quantum computing, and high-performance computing.

NAGA RAJU received the M.Sc. degree in pure mathematics from the Central University of Hyderabad, the M.Tech. degree in computer science and technology from the University of Mysore, and the Ph.D. degree in computer science and engineering from Osmania University, Hyderabad. He has been an Assistant Professor with the Department of Computer Science, Central University of Rajasthan, since May 2012. He has published around 40 international conference papers and 20 peer-reviewed journals indexed by SCIE, Scopus, and DBLP. He has established Networking Laboratory in collaboration with UGC and MeitY project funding. He is the Nodal officer for Visvesvaraya Ph.D. Scheme and Coordinator for Campus Networking Committee, Central University of Rajasthan. His research interests include wireless sensor networks, ad-hoc networks, energy-efficient scheduling algorithms in highly distributed systems, artificial intelligence, and machine learning applications in network security and network coding.

RAJANI KUMARI received the Ph.D. degree in soft computing. She is currently working with the Department of IT, JECRC University, Jaipur. She has published more than 20 research articles in refereed journals and international conferences. Her research interests include fuzzy logic, swarm intelligence, and evolutionary computing. She has worked as a Guest Editor in the *International Journal of Intelligent Information and Database System*.

ANAND NAYYAR (Senior Member, IEEE) received the Ph.D. degree in computer science from Desh Bhagat University, in 2017, in the area of wireless sensor networks. He is currently working with the Graduate School, Duy Tan University, Da Nang, Vietnam. He is a Certified Professional with more than 75 Professional certificates from CISCO, Microsoft, Oracle, Google, Beingcert, EXIN, GAQM, Cyberoam, and many more. He has published more than 300 research articles in various National and International Conferences, and International Journals (Scopus/SCI/SCIE/SSCI Indexed). He is a member of more than 50 Associations as a Senior Member and a Life Member and also acting as an ACM Distinguished Speaker. He has authored/coauthored cum Edited 25 Books of Computer Science. He is associated with more than 400 International Conferences as Program Committee/Advisory Board/Review Board member. He has two patents to his name in the area of Internet of Things and speech processing. He is currently working in the area of wireless sensor networks, MANETS, swarm intelligence, cloud computing, the Internet of Things, blockchain, machine learning, deep learning, cyber security, network simulation, and wireless communications. He was a recipient of more than 20 Awards for Teaching and Research–Young Scientist, the Best Scientist, the Young Researcher Award, the Outstanding Researcher Award, and the Indo-International Emerging Star Award (to name a few). He is acting as the Editor-in-Chief of IGI-Global, USA, journal titled *International Journal of Smart Vehicles and Smart Transportation* (IJSVST).

EKLAS HOSAIN (Senior Member, IEEE) received the B.S. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Bangladesh, in 2006, the M.S. degree in mechatronics and robotics engineering from the International Islamic University of Malaysia, Malaysia, in 2010, and the Ph.D. degree from the College of Engineering and Applied Science, University of Wisconsin Milwaukee (UWM). He has been working in the area of distributed power systems and renewable energy integration for last ten years. He has published a number of research articles and posters in this field. He is involved with several research projects on renewable energy and grid tied microgrid system at Oregon Tech, as an Assistant Professor with the Department of Electrical Engineering and Renewable Energy since 2015. His research interests include modeling, analysis, design, and control of power electronic devices, energy storage systems, renewable energy sources, integration of distributed generation systems, microgrid and smart grid applications, robotics, and advanced control systems. He is the winner of the Rising Faculty Scholar Award from the Oregon Institute of Technology for his outstanding contribution in teaching, in 2019. He is a Senior Member of the Association of Energy Engineers (AEE). He is currently serving as an Associate Editor of IEEE Access. He is working as an Associate Researcher at the Oregon Renewable Energy Center (OREC). He is a registered Professional Engineer (PE) in the state of Oregon, USA. He is also a Certified Energy Manager (CEM) and a Renewable Energy Professional (REP). He is with his dedicated research team, is looking forward to explore methods to make the electric power systems more sustainable, cost-effective and secure through extensive research and analysis on energy storage, microgrid systems, and renewable energy sources.