Received 18 October 2023, accepted 6 November 2023, date of publication 13 November 2023,
date of current version 17 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332476

RESEARCH ARTICLE

Friendship Degree and Tenth Man Strategy: A New Method for Differentiating Between Erroneous Readings and True Events in Wireless Sensor Networks

GHAIHAB HASSAN ADDAY[1,2], SHAMALA K. SUBRAMANIAM[1], (Member, IEEE), ZURIATI AHMAD ZUKARNAIN[1], (Member, IEEE), AND NORMALIA SAMIAN[1], (Member, IEEE)

[1]Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, Seri Kembangan 43400, Malaysia
[2]Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah 61004, Iraq

Corresponding authors: Ghaihab Hassan Adday (ghaihab.aday@uobasrah.edu.iq) and Shamala K. Subramaniam (shamala_ks@upm.edu.my)

This work was supported by the Universiti Putra Malaysia Contract Research under Grant 6300375.

ABSTRACT Event-driven Wireless Sensor Networks (WSNs) consist of thousands of tiny nodes. Sensor nodes are prone to faults because of their fragility and the fact that they are typically placed in harsh environments. Erroneous readings pose a high risk in many situations and affect the network's reliability, necessitating a solution to distinguish between true and faulty events. In response to this challenge, this work proposes the Friendship Degree and Tenth Man Strategy (FD-TMS) method for true event detection in WSNs. This new method can differentiate between erroneous readings and true events in a distributed manner. The FD idea has previously been used to solve security problems, while military intelligence operations have inspired the TMS and have never been used in WSNs. The FD-TMS consists of two stages. In the first stage, it employs a majority voting approach considering the friendship degree among voters. Voting among only trustworthiness nodes with high FD values will effectively differentiate true events and incorrect measurements. The second stage will validate the voting process through a novel perspective based on the TMS. TMS will check the voters' replies based on the event's location. The proposed method will delete erroneous readings, while only the true event reports will be reported. FD-TMS was comprehensively assessed in a simulation environment utilizing a performance analysis tool constructed on Java. The results were compared to the baseline algorithm, highlighting key parameters like false alarms and event detection accuracy. The simulation results demonstrated the proposed approach significantly enhanced the performance of the baseline works.

INDEX TERMS Event-driven, friendship degree, measurement faults, tenth man strategy, wireless sensor networks.

I. INTRODUCTION

The main goal of the Fourth Industrial Revolution (IR 4.0) is to create an industrial environment that is smarter and better connected, which can make industries more competitive and improve the economy [1]. There are a lot of different technologies within IR 4.0, such as cloud computing, cybersecurity, intelligent machines, modeling, and simulation [2]. The Internet of Things (IoT) and Wireless Sensor Networks (WSNs) are powerful technologies that make the environment and objects smarter. These technologies are considered as the backbone of IR 4.0 [3]. Furthermore, these technologies have been in great demand in recent years due to their capabilities. By 2025, the expected number of sensors that

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

will be deployed worldwide is projected to reach one trillion [4]. WSNs are a collection of sensor nodes that sense and detect environmental events and physical parameters in a given area. Due to their cheap cost and durability, WSNs are used in a range of applications, including agricultural, health monitoring, and military surveillance [5], [6]. Sensors are typically scattered randomly by helicopter or airplane in harsh environments. Thus, the network topology is random. In addition, sensors have a certain amount of energy. Most of the time, it is difficult to replace or change these power sources. As a result, WSNs' energy usage is a crucial concern.

WSNs can be classified into three classes based on the activity pattern: query-driven, time-driven, and event-driven [7]. Regarding these three classes, event-driven has recently gained much scientific research attention [8], [9], [10]. In event-driven WSNs, the network can sense and detect environmental events as soon as they happen. They allow the immediate detection of various events [8]. They also enable communication to be reduced and the wasteful use of node energy and computing resources to be avoided. They are often used for applications whereby it is essential to respond quickly to certain events, such as environmental observing and disaster emergencies [9].

Event detection is a crucial task in event-driven WSNs, and it is essential to prevent faulty nodes from producing erroneous readings, leading to inaccurate event reports. False events generate incorrect decisions, affect the information quality, and decrease network reliability. This work focuses on accurate event detection in event-driven WSNs and alerting Base Station (BS) while considering the possibility of sensor measurement faults. Low-cost sensor nodes are prone to malfunction, prompting serious concerns about the network's trustworthiness. Node failure may result from electromagnetic interference, power loss, and physical damage inside the sensing unit [11]. Numerous fault types, such as soft and hard faults, may lead sensor nodes to produce false data within WSN [12], [13], [14].

WSN's reliability is negatively impacted by false data reports caused by incorrect data (faulty events) [15]. If the measurement errors generated by the defective sensor are reported to BS, a wrong occurrence of a specific event will be inferred by BS. Inaccurate events that result in false reports are dangerous and pose a high risk in many applications [16]. Therefore, these faulty readings must be recognized and deleted in event-driven WSNs. Unlike erroneous readings, true readings must be reported to the BS to alert about real event occurrences (for example, a forest fire). So that BS can take countermeasures to handle the event.

Node failure is common in event-driven WSNs, with two main reasons causing it. The first reason is hardware failure, including issues with sensing units, memory, and batteries. Secondly, there are software failures, such as routing and Media Access Control (MAC) failures [12], [17]. In general, node failures result from hardware malfunction inside the sensing unit, leading to false judgments by the network due to incorrect production of incorrect data [18], [19]. Therefore, WSNs must provide advanced methods to distinguish fault events from true events [20], [21]. For example, several sensor nodes have been randomly placed in a particular area to measure physical parameters like temperature. When a sensor detects a high temperature, it does not always mean a fire has been triggered. It could be a mistake by the sensing unit of the node. If incorrect readings are sent to BS, BS will conclude that an event is happening. So, these wrong readings must be detected and removed [20].

On the other hand, nodes inside the event area can also sense unusual readings representing true natural occurrences (true events) within the environment under monitoring. These nodes must warn about a real event occurrence. These readings must be sent to the BS immediately. So that BS may take appropriate steps to deal with the problem. Distinguishing between a fault and an event is a complicated task. There are significant research efforts that have been proposed and developed to address the faulty event detection problem in WSNs [20], [21], [22], [23], [24]. One of the most well-known methods is sending all sensed data to the BS and performing a centralized analysis of the data in the BS [25], [26]. However, this approach has been analyzed to consume a lot of time and energy and cause the network to be congested increasingly. Therefore, the point where the event detection is done should be given serious consideration. Hence, distributed methods have emerged to overcome the drawbacks of the centralized methods for event detection in WSNs. Unlike centralized control, distributed (decentralized) methods, each node, backbone node, or master node is in charge of a portion of the network [27]. These normal nodes can interact directly with other nodes to perform fault detection tasks without BS interference. One of the fundamental methods in this category that has continuously evolved is based on majority voting among the node's neighbors to distinguish between real and fake events. However, using blind majority voting based on the Boyer Moor algorithm, for example, will not provide high detection accuracy and produce many faulty reports [20]. This is because many neighbor nodes may be considered faulty nodes. The following are the contributions made by this work.

- This work proposes a new algorithm designed for identifying faulty nodes. The primary objective of this algorithm is to address and minimize the problems arising from inaccurate data readings generated by faulty nodes. The technique utilizes the Friendship Degree (FD) concept to detect and exclude faulty nodes from the voting process within the network.
- This work proposes a novel algorithm inspired by the Tenth Man Strategy (TMS), a concept utilized in military organizations [28]. Rather than depending on the conventional method of blind majority voting, this algorithm checks the voters' replies from different perspectives instead of following blind majority voting. The algorithm verifies the network's topology validation to accomplish this objective, utilizing the Monte Carlo Method (MCM) to predict the event's location.

Consequently, nodes can evaluate different possibilities, thereby gaining an accurate majority voting process, resulting in a more robust and trustworthy WSN without additional equipment.

- This work performed an extensive simulation of the proposed method and compared it with existing methods in terms of performance measurement. The outcome of the proposed work has been evaluated based on the performances of the false alarm rate and event node detection, as both of these performance metrics provide trustable indications regarding the problem under investigation.

The rest of the article is organized as follows: related works on the event detection problem and previous solutions are presented in Section II. The problem formulation is explained in Section III. The proposed solution is presented in Section IV. Simulation results and the performance evaluation are presented in Section V. Section VI concludes the paper with future work.

II. RELATED WORKS

True event detection strategies for WSNs have been substantially under investigation due to their importance for the network's reliability. This section analyzes an array of existing research efforts that contributed to developing true event detection methods in WSNs.

Reference [29] developed a solution to the event disambiguation problem in the context of a sensor measurement malfunction. To identify such faults, they developed a distributed Bayesian method. The primary assumption of their proposed method is that the environmental circumstances are likely spatially correlated, and measurement mistakes caused by malfunctioning equipment tend to be uncorrelated (i.e., distinct from their neighbor nodes). In their proposed algorithm, an event decision is governed by a binary decision instead of an actual sensed reading, which makes this method more energy consumption awareness. On the other hand, in order to simplify the analysis of the Bayesian fault recognition mechanisms, the algorithm assumes that all neighbors must be inside the event region, which is the main drawback of this algorithm. Neglecting the node's voting on the event boundary will not reflect the real situation and effect on the detection accuracy. Moreover, the threshold decision scheme used in the voting process among neighbors produces incorrect decisions about the events due to the randomized nature of applying probability for the voters. The proposed algorithm is dependent on a threshold decision, so it fails to work in conditions where the environment changes rapidly. Nodes located outside and on the event's borders are forbidden from voting leading to inaccurate decisions about event occurrence.

A Localized event boundary detection algorithm based on majority voting has been suggested in [30]. This method has high performance for event detection accuracy based on the outlier detection and regional data analysis in spatial data mining. Spatial data mining procedures can extract valuable insights from sensor data. The algorithm's ability to identify pinpoint malfunctioning sensors within the network in a localized manner can be advantageous. While the algorithm focuses on event boundary detection rather than event region detection, this can still be valuable for specific tracking applications. However, instead of using binary decision, the main drawback is represented by the faulty sensor identification algorithm employed real data, which consumed and wasted energy. Energy-efficient operation is essential in sensor networks, and unnecessary information processing can quickly deplete the sensor nodes' limited power resources.

Distributed faulty sensor detection has been presented in [31]. A majority vote among neighbors allows the high detection of faulty sensors because all nodes regularly communicate the data sensed to their neighbors. However, this method required a high density of neighbors for each sensor node. Moreover, the main disadvantage of this method is that it is not energy efficient because the nodes broadcast all the sensed data to their neighbors. In addition, the proposed work required a high density of neighbors for each sensor node to accomplish the voting process successfully, which means it is unsuitable for small and medium-density networks.

Reference [32] presented a wholly distributed General Anomaly Detection (GAD) scheme for practical large-scale Networked Industrial Sensing Systems (NISSs). The proposed algorithm assumed measurement errors had Gaussian distributions. The Distributed Matching-based Grouping Algorithm (DMGA) is used to evenly distribute the division of all sensory components into tiny, highly correlated groups. Spatial correlation is a characteristic that many physical events naturally possess. Since physical events are continuous, these spatial correlations should be temporally connected to previous mappings. The proposed algorithm proves the scalability and efficiency of the event detection by computing its worst-case complexity bounds. The proposed algorithm is based only on the spatiotemporal correlations among sensors within each correlation group without any information about the event location. Moreover, the proposed algorithm suffers from high overhead and typically involves a high storage-intense operation. Thus, this approach suits industrial sensor networks composed of specific sensor nodes with high specifications.

In [33], a K-means clustering algorithm has been proposed to filter out faulty nodes using voting techniques. The algorithm can help sensor nodes to locate themselves efficiently in the presence of faults without complicated computations. However, this approach considers only the errors of the nodes' locations and neglects the sensing unit's erroneous readings that produce faulty reports. In addition, the proposed algorithm assumes the faults can happen only in the normal sensor nodes, and there is no probability for defects within cluster head nodes.

Reference [34] proposed a hybrid energy-efficient distributed clustering method to detect the true event. The cluster head nodes will be responsible for the execution of the majority voting, while the backup nodes will store all sensed data. Cluster Head nodes isolated faulty nodes efficiently by using hypothesis testing and majority voting to avoid propagating fault to higher levels. The backup nodes enhance and improve fault detection by collecting data from sensor nodes. However, this approach does not fit the homogenous networks and needs additional storage resources for backup nodes. Moreover, there is high energy consumption due to the excessive sending operation for large data packets to the backup nodes.

Reference [17] proposed a fault diagnosis algorithm using the majority of neighbors coordination. The underlying faults have been detected by belongingness using the Gaussian function. Fault readings have been detected by comparing the mean difference with standard error for different threshold conditions and timeout response for other threshold conditions, respectively. This work presented the false positive rate and the detection accuracy regarding the explanation of network size's impact on the overall approach. However, the false alarm rate is high, and the event detection accuracy is inadequate, especially with an increasing faulty percentage. This is because the proposed approach depends on the majority of neighbors coordination without considering the voters' trustworthiness.

Reference [35] proposed a Multi-Objective Deep Reinforcement Learning (MODEL) algorithm for fault detection and fault-free optimal data routing path selection. Based on deep reinforcement learning, the proposed algorithm can identify the faulty nodes with minimum energy consumption. However, consuming too much time in the learning process is the main drawback of this approach. Moreover, there is dependability on the presence of mobile sink and particular additional nodes.

Reference [36] presented a fault diagnosis algorithm based on a Deep Belief Network (DBN) to address the low detection rate problems. The algorithm classifies the faulty sensor nodes by employing a hierarchical structure of stacked multiple RBM and working through the L-by-L learning process. The proposed algorithm has low overhead and less network congestion because the algorithm does not require control message exchange between neighboring to identify the fault status. However, there is a potential high latency for the detection process and high energy consumption. This is because the true event will be detected on the BS side. Moreover, the node uses its sensed data to identify its fault status, which is not an energy-efficient approach.

Reference [20] proposed a majority voting among all neighbors based on the Boyer-Moore algorithm. This method is easy to implement without complicated computations. In addition, the proposed algorithm is energy efficient as it is a decentralized approach that uses binary decisions instead of real sensed data by neighbors. However, this approach did not provide high event detection accuracy nor a low false alarm rate. The main drawback is represented by allowing all nodes to participate in voting even though some are faulty nodes. Moreover, the Boyer Moor algorithm required many nodes to be available inside the event region, making the algorithm fail with low-density networks.

Reference [37] proposed a self-detectable distributed fault detection algorithm to detect faulty sensor nodes. Each sensor node collects data from the neighbors and then diagnoses itself using the Neyman–Pearson test and majority voting. This decentralized approach reduces BS's need for central coordination, and the Neyman–Pearson test enables sensor nodes to self-detect faults. The proposed algorithm is efficient for detecting byzantine faults such as struck-at and random faults based on majority voting among neighbors. Moreover, it is satisfactory regarding the time complexity and network lifetime. However, the proposed algorithm suffers from a high false alarm rate, especially with increasing the faulty percentage of the overall network. Thus, the main drawback is sensitivity to the network topology, which makes this approach fail when most of the neighboring sensor nodes become faulty.

Reference [38] proposed a reactive distributed fault detection (rDFD) algorithm that identifies sensor nodes having transient and permanent faults. The proposed algorithm used majority voting based on spatial and temporal correlation principles. The faulty nodes communicate with the neighboring nodes to detect their fault status according to majority voting. The main advantage of the proposed algorithm is that it requires only a small number of messages among neighbors to diagnose the faulty nodes. The computational and communication overheads have been reduced because the accuracy improvement process is exploited only if a defective node cannot detect its correct status.

Table 1 presents a range of comparisons among prior research under investigation that contributed to advancing true event detection techniques within WSNs. Even though the algorithms for differentiating the true and faulty events for WSNs have made significant scientific progress, several unresolved problems and notable concerns persist. One of the critical considerations revolves around achieving a balance between energy efficiency and detection accuracy. Many current methods often prioritize energy-saving above detection accuracy or vice versa. The scalability of the event detection algorithm is another unresolved topic. Scalability refers to the algorithm's ability to handle more sensor nodes and data without significantly decreasing performance or usefulness. Many previously presented algorithms tend to fail with high-density networks. Furthermore, some existing methods based on deep reinforcement learning algorithms have limited applications due to their time-consuming learning processes.

Despite specific techniques demonstrating high accuracy in detecting genuine events, their computational complexity and resource demand often render them unsuitable for the applications of event-driven WSNs. In addition, a wide range of existing research primarily focuses on true event detection using majority voting in a distributed way to provide valuable

TABLE 1. Comparative analysis among various related works.

Algorithm	Working Ground	Expediency	Impairments
Bayesian fault-recognition algorithm [29]	Exploited the majority voting based on the notion that faulty readings are likely to be uncorrelated.	An event is governed by a binary decision instead of real sensed data, which makes this algorithm an energy consumption awareness.	It requires a threshold decision, so it fails in fast-changing environments. In addition, outside and event-bordering nodes cannot vote, resulting in erroneous event occurrence predictions.
Localized event boundary detection algorithm [30]	Exploited localized algorithm for event boundary detection based on outlier detection and regional data analysis in spatial data mining.	The algorithm's ability to identify pinpoint malfunctioning sensors within the network in a localized manner.	High energy consumption due to usage of real sensed data in the voting process. Moreover, the proposed method worked only on event boundary detection instead of event region detection.
Distributed faulty sensor detection algorithm [31]	Exploited the regular majority voting rules for faulty sensor recognition.	The proposed algorithm is a distributed approach without BS interference, so computational overhead is low.	The main drawback is that each node regularly communicates the sensed data to its neighbors, minimizing the network lifetime. Moreover, this algorithm required a high density of neighbors for each node, which means it is unsuitable for small and medium-density networks.
Distributed General-Anomaly Detection (GAD) algorithm [32]	Exploited the graph theory and spatiotemporal correlations of physical processes to detect faulty readings.	The proposed algorithm proves the scalability and efficiency of the event detection by computing its worst-case complexity bounds.	The suggested algorithm has significant overhead and requires a considerable amount of storage.
Clustering-based DV-Hop fault-tolerant algorithm [33]	Exploited K-means clustering and majority voting techniques to filter out the faulty nodes.	The algorithm is straightforward and fast to implement because only the non-faulty nodes are used in error recognition in a centralized way.	The main drawback of the proposed algorithm that it did not consider the presence of error probability in cluster nodes.
Hybrid Energy-Efficient Distributed (HEED) algorithm [34]	Exploited the use of backup nodes and majority voting by the cluster head nodes to improve fault tolerance.	Cluster Head nodes detected and isolated faulty nodes efficiently by using hypothesis testing and majority voting.	This algorithm does not suit homogenous networks with one type of sensor node and needs additional storage resources for backup nodes. Moreover, the proposed algorithm consumes high energy due to the high transfer rate for data.
Fault diagnosis algorithm [17]	Exploited the majority voting and compared the mean difference with standard error according to different thresholds.	The faults have been detected by belongingness using the Gaussian function. This work presented high detection accuracy for many defects, such as link failure.	The false alarm rate is high, especially with an increasing faulty percentage to 20%. This is because the proposed approach depends on the majority of neighbors coordination without considering the voters' trustworthiness.
Multi Objective-Deep reinforcement Learning (MODEL) algorithm [35]	Exploited the deep reinforcement learning for detecting faulty nodes and reliable data transmission.	The proposed algorithm is efficient for fault detection with low energy consumption.	The proposed algorithm consumes much time for the learning process, and there is dependability on the presence of mobile sink and particular additional nodes called agent nodes.
Deep Belief Network (DBN) algorithm [36]	Exploited Restricted Boltzmann Machine (RBM) and the L-by-L learning process.	The proposed algorithm does not require control message exchange between neighboring to identify the fault status because the event will be detected at the BS side.	High latency and slow detection process for events due to the centralized approach. Moreover, the node uses its real sensed data to identify events, which is not an energy-efficient approach.
True Event-Driven and Fault Tolerant Routing (TED-FTR) Algorithm [20]	Exploited majority voting based on the Boyer-Moore algorithm to detect the true event.	Straightforward to implement without complicated computations. Moreover, the algorithm depends on the binary decision to conclude the event occurrence, so it is an energy-aware algorithm.	Requires the number of nodes inside the event region to be greater than those outside the event region. Thus, the proposed algorithm is suited only for high-density networks.
Distributed Byzantine fault detection algorithm [37]	Exploited Neyman–Pearson testing and majority voting among neighbors	The Neyman–Pearson test allows sensor nodes to auto-diagnose errors independently. Moreover, it is satisfactory in terms of time complexity and message complexity.	High false alarm rate and the algorithm tends to fail when most of the neighboring sensor nodes become faulty.
Reactive distributed fault detection (rDFD) [38]	Exploited majority voting based on the principle of spatial and temporal correlation regarding	The proposed algorithm requires the exchange of only a small number of messages among neighbors to diagnose the faulty nodes, so the computational and communication overheads are low.	The algorithm depended on spatial and temporal correlation threshold values, so it fails to work in conditions where the network topology changes rapidly.

information about the event [20]. However, this direction needs more attention to enhance the performance of true event detection.

The required enhancements are necessary, especially regarding high detection accuracy for faulty nodes, low false alarm rate generated from erroneous readings, message complexity, delay in the detection process, and energy consumption. Hence, considering the earlier studies' constraints using majority voting, there is a real need to propose a new, efficient, and distributed algorithm. The proposed algorithm will manage authentic event detection, focusing on delivering exclusively accurate reports to BS.

III. PROBLEM FORMULATION

Assume K is a number of sensors that are randomly deployed in a specific area. Periodically, these sensors sense and collect environmental data such as temperature. If any node K_i senses unusual readings UR_i^t during a sensing period t that is higher than a predetermined threshold (θ), the node K_i must take action to inform the BS about the event occurrence.

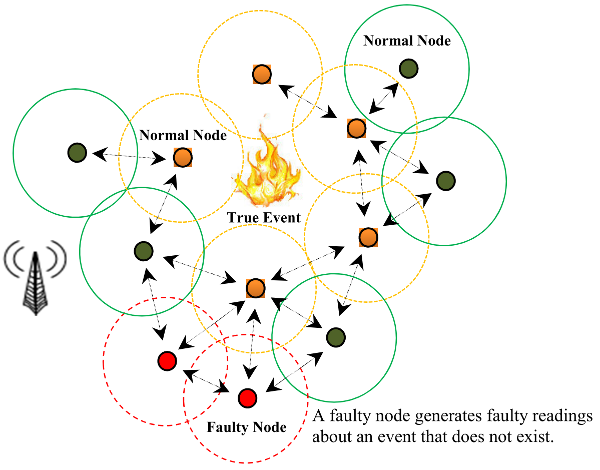
Considering the possibility of sensor measurement faults, the high sensed value may represent a high temperature in the case of a fire or an erroneous reading. It is necessary to determine whether the high sensed value is a true event or a measurement fault due to a malfunction in the sensing unit, as the possibility of sensor measurement faults is considered.

The majority voting technique based on the Boyer-Moore algorithm represents one important solution for the phenomenon under investigation [20]. Boyer-Moore algorithm is one of the famous techniques for pattern searching. Boyer-Moore algorithm can be presented for determining which of several candidates or decisions has received a majority of the votes cast in an election [39], [40]. According to this algorithm any sensor K_i that senses unusual readings will start to send request messages to all its neighbors as shown in Fig 1. Whenever K_i begins receiving replies from its neighbors, the sensor node will count the positive and negative replies. Positive replies represent that the neighbor node sensed the same unusual reading, while negative replies mean the neighbor node does not sense any readings. Suppose the sum of positive voters is greater than the positive voters. In that case, the K_i will generate a report packet and send it to the BS. Otherwise, K_i concludes there is no event occurrence and schedules the next sensing round.

Obviously, this blind majority algorithm does not consider any information about the trustworthiness of the neighbor nodes, as many of them can be faulty nodes, and their voting negatively affects the voting process. Moreover, the Boyer Moor algorithm does not include any information about the network topology and event location. This means the request messages will be sent to all node's neighbors, even those that are out of the event boundaries. Sensor nodes out of the event range will also negatively affect the voting process. If K_i produces an incorrect reading, BS will incorrectly assume that an event has occurred, leading high false alarm rate, low detection accuracy, high energy consumption, network congestion, and loss of the network's reliability.

As a result, measurement errors must be discovered and removed as soon as possible. To differentiate between a real event and a fault event, this work used a new majority voting algorithm among adjacent nodes. True events are transmitted to BS via the geographic routing algorithm proposed in the baseline work [20], while faulty readings are disregarded in a decentralized way by sensor nodes without interference by the BS.

Definition 1: A true event is an environmental occurrence, such as a fire in the forest or specific liquid flooding, that raises the value of the sensor parameter being measured over its typical range [20].



A faulty node generates faulty readings about an event that does not exist.

🟩 Normal sensor nodes residing outside the event region without any sensed value.

🟨 Normal sensor node residing near the event region, producing accurate information about true event occurrence.

🟥 Faulty node residing outside the event region produces inaccurate information about a nonexistent event.

FIGURE 1. Problem definition illustration.

Definition 2: A measurement fault occurs when a sensor's hardware fault causes the measured parameter to be miscalculated due to malfunction inside the sensing unit.

A. RESEARCH ASSUMPTIONS

This study investigates variables that may affect the sensor faulty reading phenomenon, which is under investigation. Thus, examining particular correlations by establishing well-defined assumptions will help to comprehend the problem. Following are the research assumptions that guide the inquiry and help make meaningful conclusions from the evidence.

- Each node has a fixed position.
- The network has only one BS.
- All nodes, including BS, have the same restricted transmission range D.
- All nodes except BS have limited battery capacity.
- All nodes are aware of their location and remaining energy.
- All nodes sense the environment parameters periodically.
- There is a probability of erroneous sensor measurements.

B. NETWORK MODEL

In the proposed network model, K is a number of sensor nodes, K_i ($i = 0,1,2, 3,\ldots, i\text{-}1$), that are randomly scattered in a Two-Dimensional *2D* geographical area with dimensions of (X, Y). Any two nodes, such as K_i and K_j can communicate directly if the Euclidean Distance $ED_{i,j}$ between them is less than or equal to their transmission range D. Thus, neighbor nodes are two nodes that can communicate directly. Assume that (X_i, Y_i) and (X_j, Y_j) are the coordinates of nodes K_i and K_j, respectively. Equation (1) is used to calculate the Euclidean Distance between any two neighboring nodes.

$$ED_{ij} = \sqrt{(X_j - X_i)^2 + \left(Y_j - Y_i\right)^2} \tag{1}$$

As seen in Fig. 2, any node not neighbor of the BS will choose one of its neighbors as the forwarding node to transmit its sensed data to the BS. The proposed work used the geographic routing algorithm of [20].

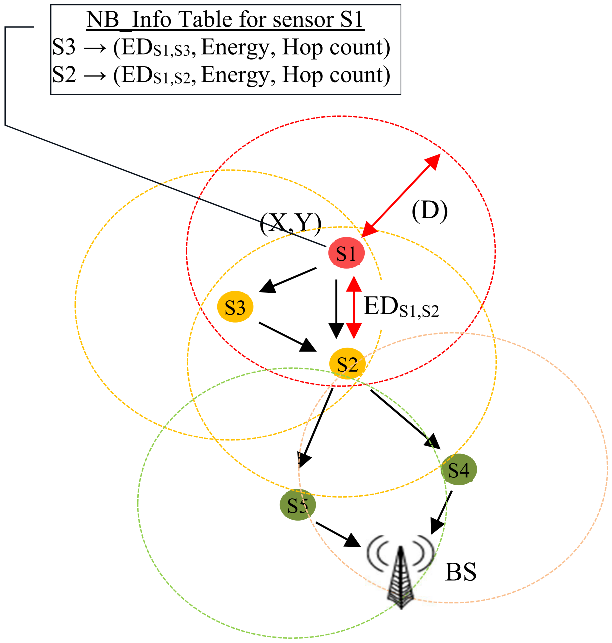

NB_Info Table for sensor S1
S3 → (EDS1,S3, Energy, Hop count)
S2 → (EDS1,S2, Energy, Hop count)
(X,Y)
(D)
S1
S3
EDS1,S2
S2
S4
S5
BS

FIGURE 2. Network model.

C. EVENT MODEL

Any environmental occurrence, like a fire in a forest, takes up a sizable ground area in whatever shape. A variety of environmental conditions, such as wind, can influence event area and event shape. Without losing the generality and for simulation requirements, this work has presented the event area as a circular sector with a fixed center *(X, Y)* and fixed radius R as shown in Fig. 3.

Periodically, nodes that are equipped with sensors sense physical parameter value under monitoring and decide whether it exceeds a predefined threshold (θ). The proposed work has considered measurement faults. Therefore, a high sensed value may also be a faulty measurement by nodes residing outside an event region. Using the proposed algorithm for true event detection, the sensor nodes can determine whether the sensed value is a true event or an erroneous reading. The true event will be delivered to the BS using the routing algorithm, and the erroneous reading will be ignored. The event detection algorithm verifies whether the sensed value exceeds the threshold and whether it is a mistaken measurement or a true event. The node disregards false readings while the routing algorithm relays only actual events to the BS in a multi-hop communication manner.

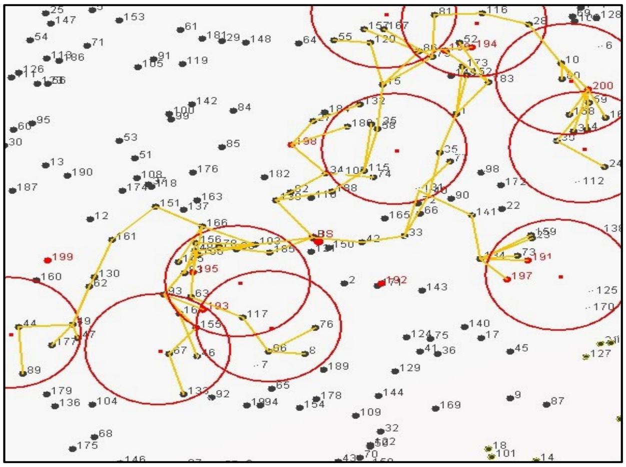


FIGURE 3. Random network topology with ten events scattered randomly.

IV. THE PROPOSED EVENT DETECTION METHOD

The majority voting technique based on the Boyer-Moore algorithm has two main drawbacks. First, faulty nodes that can participate in the voting process may affect gaining an accurate decision about the event and produce a high false alarm rate. This is due to the algorithm's behavior that makes every node broadcast a Request message asking about event occurrence to all neighbors in its communication range.

According to the Boyer Moor algorithm, a true event occurs if the positive replies are greater than or equal to the negative ones. Otherwise, there is no event, and it is false reading. Thus, there is no consideration of whether the voters are faulty or trustable nodes. Moreover, the faulty nodes always flood the network with Request messages asking about an event that does not exist. The second drawback of the majority voting technique based on the Boyer-Moore algorithm is the absence of any information about the event location that randomly occurs anywhere inside the network topology. Not including the event location factor will allow all neighbors to apply for voting, even if some of them are outside the event borders. This also negatively affects the last decision about true event occurrence and the network's reliability.

This work presents two techniques that can work together for true event detection in event-driven WSNs. One of them will recognize the fault nodes and prevent them from participating in the voting, while the second one will manage the voting process for the nodes outside the event borders. Friendship Degree and Tenth Man Strategy (FD-TMS) will be able to make sensor nodes detect the true event occurrence. On the other hand, the sensor nodes will discard the faulty event in a decentralized way without BS interference. A detailed description of the overall method entitled FD-TMS for true event detection follows.

A. FRIENDSHIP DEGREE (FD)TECHNIQUE

Many previous approaches based on the voting idea used several different voting kinds, such as absolute majority, plurality majority, or Boyer Moore majority voting [20], [37], [41]. The primary basis for the majority voting concept is the actual sensed data tend to be spatially correlated, while faulty readings are stochastically uncorrelated. Therefore, absolute trust in the neighbor's votes is essential to these schemes. Constructing a fault management framework based on one metric is insufficient and leads to minimizing the event detection accuracy.

In many situations, when a sensor node senses unusual reading, it will broadcast a request to all its neighbors and wait for a reply. The replies of the faulty nodes do not reflect the real situation inside the environment because they are already faulty nodes due to malfunctioning sensing units for any reason. Such scenarios dramatically minimize the network's reliability and do not reflect the actual situation inside the environment. Furthermore, the network's inability to deliver specific and correct information to the BS may pose a high risk, especially in monitoring applications such as medical applications and forest fire monitoring applications.

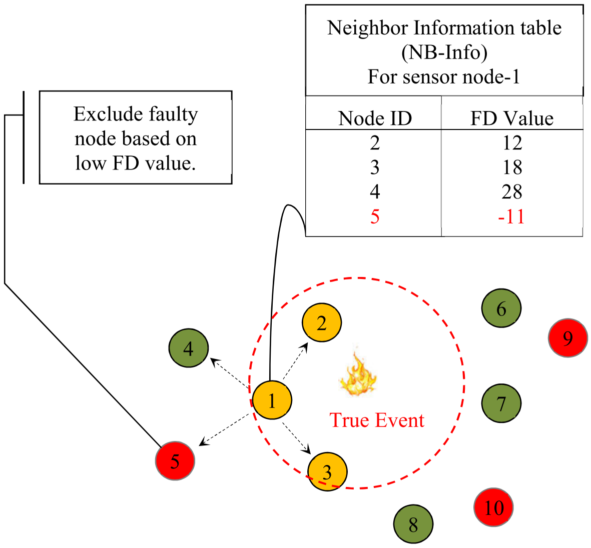
This work presented an FD algorithm that can overcome the shortcomings of traditional majority voting. FD creates a new concept in event-driven WSN inspired by daily human life. A similar idea was used previously to solve some security problems in WSNs. However, this is the first time using this technique for event detection. The basic idea is that when any node senses unusual reading, it will build the event decision based on the nodes with the highest FD values. FD represents the previous history of the neighbor's behavior in producing faulty readings. FD value for any node is updated continuously by exchanging Requests and Replies messages during the sensing period. Trustable nodes will get a high value of FD due to their accurate readings. Unlike trustable nodes, faulty nodes will have low FD values.

The proposed algorithm will initialize FD as a zero value for all sensor nodes at the beginning. At the end of every sensing period, all nodes will update the FD for their neighbors according to the final decision that it made about event occurrence. Step by step, with an increasing number of sensing rounds, the trustworthiness of non-faulty nodes can be achieved and will be updated to have high FD values. While the faulty nodes will have low FD values due to continuous behavior in generating faults. The proposed approach will exclude the faulty nodes from participating in the voting process based on their FD values, as shown in Fig. 4. Any node that has an FD value less than the predetermined threshold will be neglected during the majority voting process. The predetermined threshold for the low FD value was assigned to value -3 to gain more accurate results for true event detection (See Algorithm 2). Thus, more accurate decisions regarding the event disambiguation operation will be considered. In addition, the proposed FD will preserve energy because it will reduce the sending operations of useless information and false report data. Moreover, the network will not be highly congested with control packets.

The following scenario is an illustration of the FD technique. At the beginning, all nodes will have an FD value equal to 0, and if any node K_i during the sensing period t senses unusual reading UR_i^t more significant than the (θ), it will broadcast a request to its neighbors K_j (j = 0, 1,2,3,…, j-1), asking if they sense the same reading. Environmental occurrences (real reading) are spatially correlated. That means K_i senses unusual readings comparable to its neighbors, while measurement mistakes caused by malfunctioning equipment are likely to be uncorrelated and differ from its neighbor's readings [13]. After K_i receives the replies, it applies the second strategy TMS to make accurate decisions about event detection (See Section IV-B). If the event is confirmed, K_i will update its FD_i and its neighbor's FD_j (maximizing the FD for all neighbors who replied with one and vice versa).

On the other hand, the neighbor K_j,who receives a request from node K_i, will also update the FD value for itself and others based on the sensed value and the event confirmation. In this way, the FD column inside the Neighbor Information table (NB-Info) will be more accurate with time advances. FD value for a faulty node may decrease to −30, while the trustable nodes may reach an FD of more than 20.

The proposed method will not only recognize the faulty nodes inside the network, but it will also make the faulty nodes identify themselves as faulty nodes. FD technique will allow only the trusted nodes to participate in voting, preventing the faulty nodes from voting and causing chaos. This will help to get high event detection accuracy and minimize the false alarm rate because the majority voting will not be based on a general perspective.



Neighbor Information table (NB-Info) For sensor node-1

Node ID	FD Value
2	12
3	18
4	28
5	-11

FIGURE 4. Excluding the faulty nodes from participating in the voting process.

B. TENTH MAN STRATEGY (TMS) AND TOPOLOGY VALIDATION

As mentioned above, the FD technique (the first stage in the proposed method) will allow us to recognize the faulty nodes as troublemakers. However, another shortcoming needs to be addressed in the blind majority voting or majority voting based on the Boyer Moor algorithm in previous research works. Any sensor node that senses an environmental parameter must ask all its neighbors, and many of these neighbors are not faulty nodes, and they reply correctly. However, their location outside the event border will negatively affect the voting process due to sensing range limitations.

Moreover, one of the most challenging cases is when a faulty node is located near the event borders. In this case, the usual voting will tend to fail because many neighbor nodes (inside the event region) of this faulty node will confirm event occurrence. However, no real event is sensed by the faulty node; faulty nodes always produce incorrect readings in every sensing period. FD technique alone cannot deal with this problem because many voter nodes are highly trustworthy. Therefore, another crucial factor that was not considered in previous works on event detection problems is network topology validation, which is significant during voting and must be included. This work proposed the use of the TMS, which includes the topology validation factor, to deal with voters located outside the event borders.

Many intelligence agencies worldwide have used the TMS to collect vital and accurate information about neighboring and enemy countries. TMS was used to challenge conventional and received information to consider alternative directions. TMS says if nine people agree on a specific opinion, the tenth person must take a contrary approach. The proposed method will mimic this concept in the events differentiating process. To the best of our knowledge, this is the first time using this strategy in WSNs.

The proposed TMS and topology validation will make the node take a period of suspicion to check the voters' responses according to the location of the neighbors and the expected event location. This way, the sensor node will know the opinion of its neighbors about it. Network topology validation refers to the procedure by which sensor nodes determine and predict an event's location and identify the neighboring nodes capable of sensing that event based on their proximity.

The main steps of modeling the TMS and topology validation concept consist of using the MCM for the topology validation process. The MCM is a computational algorithm that uses repeated random sampling to obtain numerical results. It is simple, fast, and does not require complex computational resources. This proposed work used MCM as a statistical technique to estimate the event's location and compile a list of neighboring nodes that can potentially detect that event. Using MCM, each sensor node creates an Expected Binary String (EBS) (binary string with zeros and ones) for each event based on the distance from the neighbor to the expected event location. The assigns a binary decision, represented as 0 or 1, to indicate its expected ability to sense the event. A value of 1 signifies that the specific neighbor is within the event range and capable of sensing the event, while a value of 0 indicates that the neighbor is situated far from the event region and incapable of sensing the event. It is imperative to execute the topology validation before the TMS can effectively address the issue of nodes positioned outside the event region. Importantly, this process occurs in conjunction with finalizing the first stage, which involves the FD algorithm and the majority voting phase.

On the other hand, TMS will take action to check the voter's replies according to the topology validation procedure so the final decision of the event occurrence can be made accurately. When the sensing period starts, any node sensed unusual readings will finish the Request sending and Replies receiving. During this time, the sensor node will also complete the initialization of the FD stage and start generating the Collaborative Binary Decision (CBD) based on the voters' replay. CBD reflects the collective judgment of neighboring nodes regarding a particular event. CBD is a binary decision, where a value of 1 indicates the event occurrence, while 0 shows there is no event. This judgment is made by comparing the sensed data to a predefined threshold (θ). FD will relieve CBD from the faulty nodes voting, but we must execute the TMS to solve the problem of the nodes outside the event region. The proposed TMS will accurately conclude the event occurrence by comparing the CBD with the EBS stored previously from the MCM. TMS will match and count the number of positive replies to perform the proposed majority voting according to the sensed value and the topology validation.

FD and TMS algorithms are essential for precise and clear majority voting. Both algorithms will gain high accuracy for event detection and the lowest false alarm rate. Moreover, the proposed method will preserve energy consumption and minimize network congestion. Below is a detailed explanation of the proposed event detection method that follows a step-by-step process, and the pseudocode representation of our event detection method is given in the FD-TMS method.

- In the beginning, BS will start broadcasting Hello messages to nodes within its translation range. Hello message was broadcasted only once time, containing node_id, pos_x, pos_y, hop count, and remaining energy [20]. This message is used to share the geographic location of each sensor node. Whenever any node Ki received a Hello message from its neighbor K_j. K_i will compute and store Euclidean Distance ED_{ij} using Equation 1. in the NB_Info table. This table will contain all the needed information for the proposed true event detection algorithm, such as distance among neighbors and the FD value for each neighbor node. Meanwhile, FD values for sensor nodes will be updated periodically by the exchange of the Request and Reply messages during the voting process. Algorithm 1 illustrates pseudocode for the initialization stage for each sensor node.

Algorithm 1 Initialization Stage for Any Node K_i

```
Function Initialization ()
    If $K_i$ is BS Then
        Broadcast (Hello Message);
End
Function Received Hello ()
    If $K_i$ receive Hello form a neighbor $K_j$
        Compute $ED_{ij}$ using Equation 2
        Store Node_ID;
        Store ($pos\_x$, $pos\_y$);
        Store $ED_{ij}$ in NB-Info Table;
    End
End
Function Send Hello ()
    After receiving Hello message from any node $K_j$
    Foreach $K_j$ in NB_Info Table Do
        Broadcast (Hello);
    End
End
```

- For each sensing period t, any node K_i senses the environment, compares the current perceived value with a predetermined threshold (θ), and creates a binary decision BD_i^t. If the sensed value SV_i^t over the period t is greater than (θ), the node K_i makes a binary decision BD_i^t as 1; otherwise, zero. If BD_i^t is one, the node sends a request message to its neighbors and waits for their replies as described in (2). Algorithm 1 illustrates pseudocode for the sensing function for each sensor node.

$$BD_i^t = \begin{cases} 1 & \textit{if } SV_i^t > \theta \\ 0 & \textit{Otherwise} \end{cases} \tag{2}$$

Algorithm 2 Sensing Function for Any Node K_i

```
Function DataSensing ()
    If sensingTimer () notStarted Then
            Start sesingTimer();
    Reset NB_Event Table;
    If Sensed value $\geq \phi$ Then
        $Bd_i = 1$;
        Forall $K_j$ in NB_ Info Table Do
            Broadcast(request);
        End
    Else
        $Bd_i = 0$;
    End
End
```

- The node K_i will receive replies from its neighbors K_j and check the probability of recognizing itself as a faulty node. If the FD_i of the node is lesser than the predetermined threshold, the TMS function will be activated. This way, the node will think differently based on whether the neighbors see it inside the event region. Otherwise, K_i will follow the typical event checking, executing the straightforward majority voting and executing topology validation. Both EventChecking_TMS and Normal EventChecking functions will be explained separately in Algorithms 4 and 5, respectively. Algorithm 3 illustrates pseudocode for data receiving function within the proposed work.

Algorithm 3 Receiving Data Function for Any Node K_i

```
Function ReceivedData(Data)
    If Data.type = Reply Then
        Put Data.node_id, Data.Bd and $InsideEvent\_Border$
        in NB_Event;
        If count (Reply) >= 3 Then
            If $FD_i$ <= Threshold_2 Then
                Start Event_Checking_TMS
            Else
                Start Normal Event-Checking (without TMS)
            End
    If Data.type = Request Then
        If $Bd_i$ =1 Then
            If Euclidean Distance (Expected Event Location,
            $K_j$ Location) <= 70 Then
                $InsideEvent\_Border_j = 1$;
            Else
                $InsideEvent\_Border_j = 0$;
            End
        Data_S.Bd = $Bd_i$;
        Data_S.InsideEvent_Border = $InsideEvent\_Border_i$
        Data_S.fromNode_id = Node id;
        Data_S.toNode_id = data.Node id;
        Data_S.type = Reply;
        Send Ddata_S to Data.Node id;
End
```

- In the TMS function, K_i will check neighbors' replies and watch their Bd_j and $InsideEvent\_Border_j$ values. The InsideEvent_Border parameter represents the neighbor's opinion regarding the event's and node's locations based on the Euclidean distance. Positive confirmation for both parameters represents that the neighbor node inside the event border and detected the event's readings will add 1 to the CBD, as shown in (3). Algorithm 4 illustrates pseudocode for event checking using TMS.

$$Event_i^t = \begin{cases} Add\ 1\ to\ CBD & \textit{if } BD_j \textit{ and } Inside\_Ev_j = 1 \\ Add\ 0\ to\ CBD & \textit{Otherwise} \end{cases} \tag{3}$$

- Assuming the node K_i has a high FD_i value, there is no need to use TMS based on the high trustworthiness of the node itself. In this case, K_i activates the normal event-checking function. Nodes with high FD will not broadcast request messages to their neighbors unless they are actually inside the event region and sense

Algorithm 4 EventChecking_TMS Function for Any Faulty Node K_i With Low FD Value

```
Function EventChecking_TMS ()
    Forall $K_j$ in NB-Event table do
        If $Bd_j = 1$ AND $InsideEvent\text{-}Border_j = 1$
            Add 1 to the CBD;
        Else
            Add 0 to the CBD;
        End
    End
Call Function Event Decision ()
END
```

unusual readings. However, our proposed normal event checking differs from other majority voting approaches because it will neglect the votes of the faulty nodes using their FD information representing the previous historical behavior in generating false readings, as shown in (4). Algorithm 4 illustrates pseudocode for normal event checking where CBD represents the decision made by the majority voting process while BD_j represents the binary decision made by the neighbor node.

$$Event_i^t = \begin{cases} Add\ 1\ to\ CBD & if\ BD_j = 1 \\ Add\ 0\ to\ CBD & Otherwise \end{cases} \tag{4}$$

Algorithm 5 Normal EventChecking Function for Any Normal Node K_i With High FD Value

```
Function EventChecking()
    Forall $K_j$ in NB-Event table Do
        If $Bd_j = 1$ Then
            Add 1 to the CBD;
        Else
            Add 0 to the CBD;
        End
    Call Event Decision ();
End
```

- K_i Will match the CBD with the EBS from the MCM using the idea of topology validation and update the FD value for itself and its neighbors in the NB_Info table as shown in (5), and Algorithm 5, which illustrates pseudocode for Event Decision function.

$$Event_i^t = \begin{cases} Report & if\ CBD = EBS \\ No\ Event & Otherwise \end{cases} \tag{5}$$

- For the MCM operation used in the proposed algorithm, every sensor node executes MCM only once during the network initialization phase. Based on the simulation experiences for the proposed work, what should be mentioned here is that the maximum EBS number will not exceed 12 binary strings because our method will exclude the duplicated binary string. Algorithm 6 illustrates pseudocode for MCM operations.

Algorithm 6 Event Decision Function for Any Node K_i

```
Function Event Decision ()
    Forall Expected Binary String (EBS-List)
        If the CBD = Any EBS inside EBS-List Then
                Data.type = Report;
                Data.node_id = $K_i$;
                Data.Bd = $Bd_i$;
                Send Data to Next-hop;
                Forall $K_j$ in NB-Event table Do
                    If $Bd_j$ =1 Then // Update $FD_j$
                        $FD_j$++
                    Else
                        $FD_j$ –;
                    End
                    $FD_i$++; //Update my $FD_i$
                End
            Else
                $FD_i$ –; //Update My $FD_i$
                If $Bd_j$ =1 Then //update $FD_j$ neighbors
                    $FD_j$ –;
                Else
                    $FD_j$++;
                End
        End
    End
End
```

Algorithm 7 Monte Carlo Method for any Node K_i

```
Function Monte Carlo Method ()
    Define $K_i$ Transmission Range boundaries;
    Set $Z$ number of random iterations;
    Generate random expected event location;
    Forall $Z_i$ sample DO:
        If $K_i$ inside the expected event region Then
                Add 1 to EBS;
        Else
                Add 0 to EBS;
        End
    End
    IF EBS is Not in EBS_List Then
        Store EBS in the EBS_List
    Store the Expected Event Location
End
```

C. COSTS ANALYZATION

In the following sub-parts, we examine our scheme's computational and communication overhead costs.

1) COMPUTATIONAL COSTS

Calculating the computational cost of our proposed method involved two stages, which are (1) the FD-TMS cost and (2) the MCM cost. In the first phase regarding the true event detection, the majority voting based on the FD-TMS method requires O(n) time complexity. On the other hand, the MCM plays a vital role in the proposed work to predict the event's location. The number of simulation samples N affects the computational complexity of the MCM [42], [43]. The MCM executes a fixed number of independent random trials in the proposed algorithm regardless of input size or problem complexity. In that case, its time complexity can be expressed as O(n), where n is the number of neighbors. Therefore, the entire proposed method has an O(n) total time complexity.

2) COMMUNICATION COSTS

In the proposed event detection method, any node K_i during sensing unusual readings, will broadcast Request messages to all its neighbors. On the other hand, all neighbors will respond with Replay messages. If we assume the average number of neighbors for each node is *AVG*. Then, the differentiation process between true and faulty events will require $(1+AVG)$ messages. In the same context, for any sensing period (t), if there is a number of faulty nodes Q and a number of event area nodes W, there will be a total of $(Q + W)(1+AVG)$ number of transmission messages required. Moreover, W needs to send report packets via a multi-hop communication style. We assume the average hop count toward BS is H. Thus, WH will be the number of nodes participating in sending reports toward BS. Therefore, the total number of messages for a single detection round is given by (6).

$$Dete_m = (1 + AVG)(Q + W) + WH \qquad (6)$$

V. PERFORMANCE ANALYSIS

The following subsections present a clear analysis of the simulation configurations, parameters, and results analysis that will be illustrated in detail for the proposed true event detection method.

A. SIMULATION SETUP AND PARAMETERS

We designed and developed a new WSN simulation tool to evaluate the proposed method and benchmark work. The proposed simulator took advantage of the power of Discrete Event Simulation (DES) and was entirely built in Java Eclipse. This simulation tool has been carefully made to meet the WSN study and testing needs. By developing our performance analysis tool, we can change every aspect of the modeling process. This allows us to configure settings, add custom methods, and try different network scenarios. We ensured that the simulator had an easy-to-use interface by taking advantage of Java programming language's flexibility.

The developed custom-built simulation platform for this research can become a trustworthy tool for studying how WSNs behave, improving routing protocols, and testing different fault tolerance frameworks. This work has chosen the True Event-Driven and Fault Tolerant Routing (TED-FTR) algorithm as a baseline algorithm [20]. The routing algorithm will be used to forward the event reports to the BS via a multi-hop communication style. For the simulation process, 100 to 1000 nodes were scattered randomly in an area (600,600) with a step length of 100. The BS is located in the center of the region (300,300) as shown in Fig. 5.

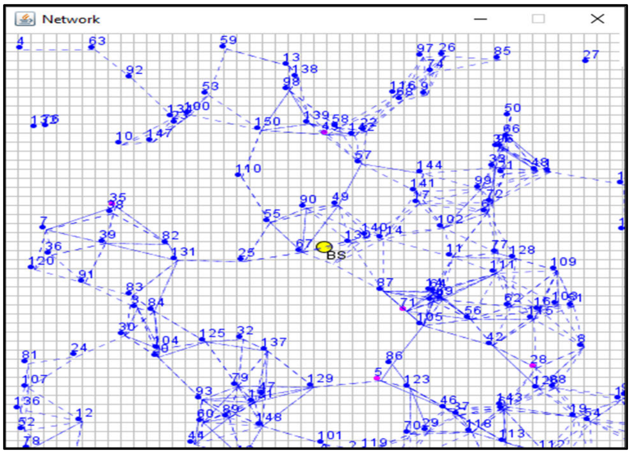


FIGURE 5. Random network topology for 150 sensor nodes with BS at the middle of the network.

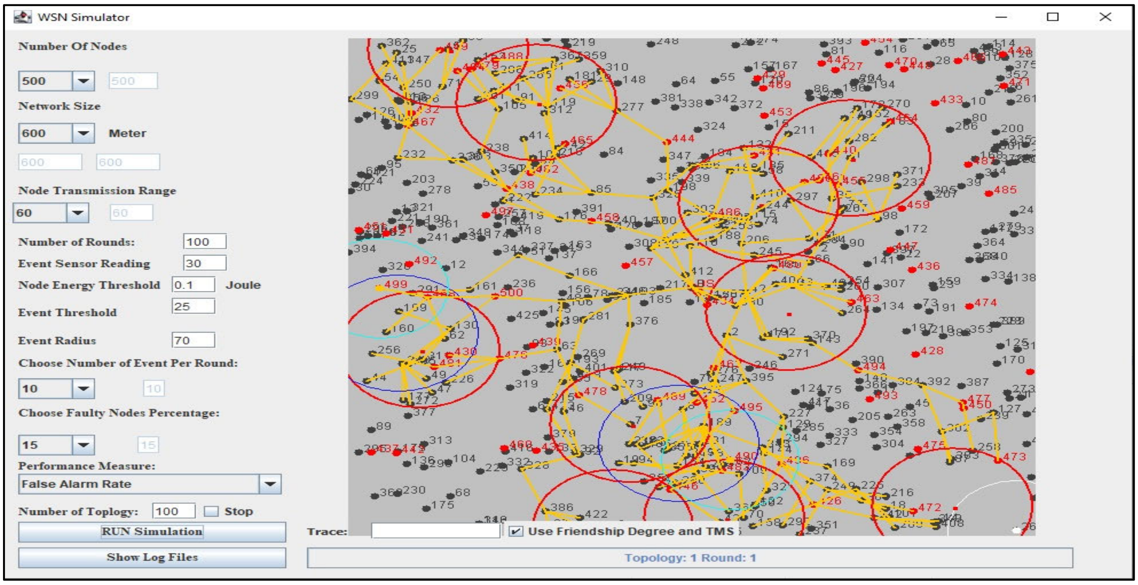
All nodes have the same transmission range of 60 m. Normal sense readings are drawn from $S\ (m_1, q_1^2)$ and real event readings from $S\ (m_2, q_2^2)$. Where $m_1 = 10, m_2 = 30$ and $q_1 = q_2 = 1$. Faulty readings are also drawn from *S(30,1)*. The (ϕ) threshold can be selected randomly between (m_1+q_1) and (m_2-q_2). The proposed work shoes $(\phi = 25)$ in our simulation. We have taken average simulation results of 100 different random network topologies. Moreover, we have taken an average of 100 different rounds with ten different random event locations for every single topology. Table 2 has a list of the rest of the simulation parameters.

TABLE 2. Simulation parameters.

Simulation Parameters	Value
Network size	600 m × 600 m
Location of the BS	300 m × 300 m
Number of nodes	100,200,300,400,500,600,700, 800,900,1000
Event radius	70 m
Transmission range	60 m
Number of events per round	10 events
Normal sensor reading	S (10,1)
True event sensor reading	S (30,1)
Faulty sensor reading	S (30,1)
Event threshold	25
Percentage of faulty nodes	5%, 10%, 15%, 20%
Initial energy for each node	15- rand (0,1) × 102 J
Energy threshold	0.1 J

B. PERFORMANCE METRICS

Most famous performance measures were used to compare the proposed method to the current algorithms. False Alarm Rate and Event Node Detection Accuracy are the main metrics that provide clear indications of network reliability while considering the probability of faulty readings occurrence [20], [44].



WSN Simulator

Number Of Nodes: 500 (500)

Network Size: 600 Meter (600, 600)

Node Transmission Range: 60 (60)

Number of Rounds: 100

Event Sensor Reading: 30

Node Energy Threshold: 0.1 Joule

Event Threshold: 25

Event Radius: 70

Choose Number of Event Per Round: 10 (10)

Choose Faulty Nodes Percentage: 15 (15)

Performance Measure: False Alarm Rate

Number of Toplogy: 100 □ Stop

RUN Simulation

Show Log Files

Trace: ☑ Use Friendship Degree and TMS

Topology: 1 Round: 1

FIGURE 6. Simulation for 500 sensor nodes with a 15% faulty percentage.

1) FALSE ALARM RATE (FAR)

FAR is a vital performance measure that assesses the network's rate of false alarms. FAR can be defined as the ratio of faulty nodes that reported a faulty event to the BS to the total number of faulty nodes within the network [20]. This performance metric evaluates the reliability and accuracy of the WSN and can be computed using (7). A lower value for FAR represents a more reliable and efficient network.

$$FAR = \frac{Number\ of\ faulty\ node\ report\ event\ to\ BS}{Total\ number\ of\ faulty\ nodes\ in\ network} \tag{7}$$

2) EVENT NODE DETECTION ACCURACY (ENDA)

ENDA can be defined as the ratio of the event area nodes that confirm events to the total number of event area nodes. Event area nodes mean all nodes that reside inside event region borders [20]. ENDA is a vital performance metric computed using (8) in the developed simulation experiments. A high ENDA means that the WSN can precisely detect true events, which is essential for gaining network reliability. On the other hand, a lower percentage of ENDA shows that the network might not be able to identify the true events correctly. This could make it hard to trust the network's performance.

$$ENDA = \frac{Number\ of\ event\ area\ nodes\ confirms\ event}{Total\ number\ of\ event\ area\ nodes} \tag{8}$$

C. RESULTS ANALYSIS

This work used different network scenarios to evaluate the proposed true event detection algorithm following the parameters shown in Table 2. For example, a simulation for 500 sensor nodes and 15% faulty nodes in our simulation tool is shown in Fig. 6. In every scenario, the proposed algorithm calculates the FAR and ENDA. FAR will be measured based on the number of faulty reports successfully reaching the BS. A low False Alarm Rate means the network's reliability can be achieved. At the same time, a high False Alarm Rate indicates the network has produced high erroneous readings, and there is a high rate of unreliable data within the network. In contrast, ENDA was measured based on the nodes inside the event region that detected the event correctly. High ENDA is required to satisfy the network's reliability, while low ENDA represents the network's inability to detect the phenomenon under monitoring. Both performance metrics represent indications of the proposed method's performance and have been compared extensively to the baseline work [20].

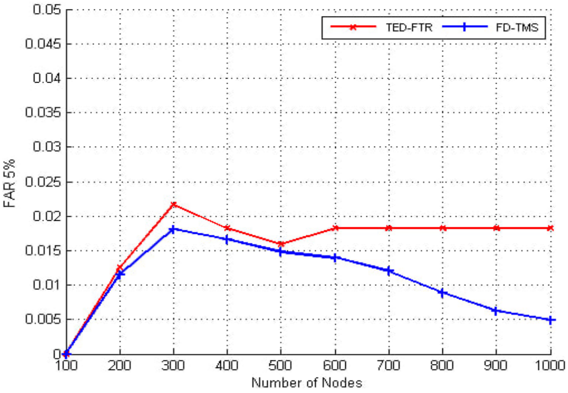


FIGURE 7. FAR for 5% faulty percentage.

1) FAR

Fig. 7, Fig. 8, Fig. 9, and Fig. 10 illustrate FAR for the suggested FD-TMS method compared to the baseline work TED-FTR algorithm [20]. The simulation analysis employed varied numbers of nodes ranging from 100 to 1000 with a step length of 100 and fault percentages ranging from 5% to 20% with a step length of 5% for a clear and complete evaluation. Fig. 7 shows the FAR with 5% faulty sensor nodes for the baseline and proposed FD_TMS algorithms. It is clear for a lesser faulty percentage, the false alarm rate for both algorithms is low. However, the proposed algorithm achieves lower FAR as the number of nodes increases. This is because the proposed FD_TMS will recognize the faulty nodes smoothly by participating more neighbors in the voting process. Moreover, the TMS will neglect the voting of the sensor nodes far from the event region. This leads to minimizing the FAR and discarding the faulty reading inside the network caused by faulty nodes. In addition, Fig. 7 shows that there is a higher FAR within 200 and 300 sensor nodes for the proposed work compared with other networks containing more nodes. As we considered the event area a fixed circle, faulty nodes close to the event borders may conclude the event from the majority voting process before being recognized as faulty nodes by the network. However, the proposed method dramatically improved the benchmark work in all situations.

Fig. 8 shows the FAR with 10% faulty sensor nodes for the baseline and proposed FD_TMS algorithms. The FAR value for the baseline algorithm is wobbling because it is based on blind majority voting.

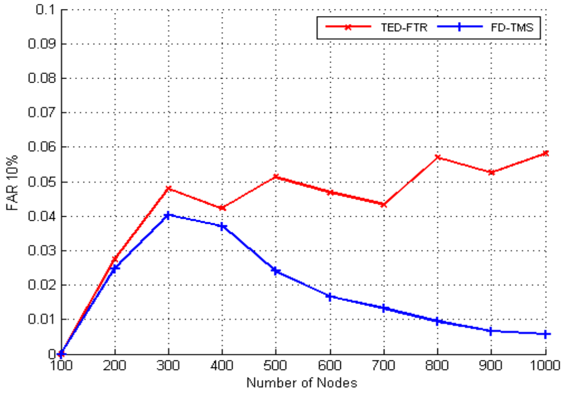


FIGURE 8. FAR for 10% faulty percentage.

The Boyer Moor algorithm tends to fail as it neglects any information about the event's location or the geographic location of the neighbor sensor nodes. Moreover, the FAR values increase with increasing percentage of faults. On the other hand, the proposed algorithm based on the new concepts of the friendship degree and the tenth-man strategy presented a lower FAR compared with previous works.

Simulation results demonstrate that the proposed method provided high efficiency and adaptability. It can gradually reduce the FAR as the network's density increases, indicating its ability to handle diverse scenarios and maintain high trustworthiness. Moreover, the proposed work presented a stable performance with an increasing percentage of faulty nodes. Unlike our proposed method, the majority voting based on the Boyer Moor algorithm showed inconsistent performance because it neglected the information about the voters and the event's location.

Fig. 9 offers insights into the performance of the true event detection process by presenting the FAR when 15% of the sensor nodes in the network are considered faulty. The baseline algorithm's FAR is noticeably higher, generating false data reports. This major concern implies that the previous algorithm tends to pass the erroneous readings to the BS. The graph emphasizes that the proposed work succeeds in decreasing the FAR gradually once the network gets more density without any consideration of the faulty percentage. Recognizing the faulty nodes and preventing them from applying for voting alongside employing the tenth-man strategy that verifies the voting process will significantly reduce the FAR to reach an ideal level.

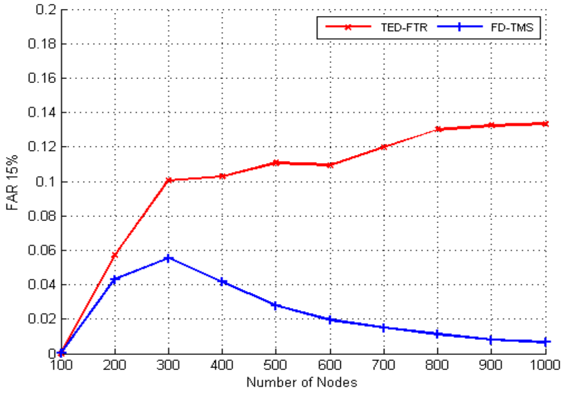


FIGURE 9. FAR for 15% faulty percentage.

Fig. 10 illustrates that the proposed algorithm outperformed the baseline algorithm by 30% when dealing with 20% malfunctioning sensor nodes. The majority voting based on the Boyer Moor algorithm presents high FAR increased significantly as the number of faulty nodes within the network increased. FAR increased dramatically with the network's increased number of sensor nodes. In contrast, the proposed FD_TMS gets lower FAR as sensor nodes increase. This is

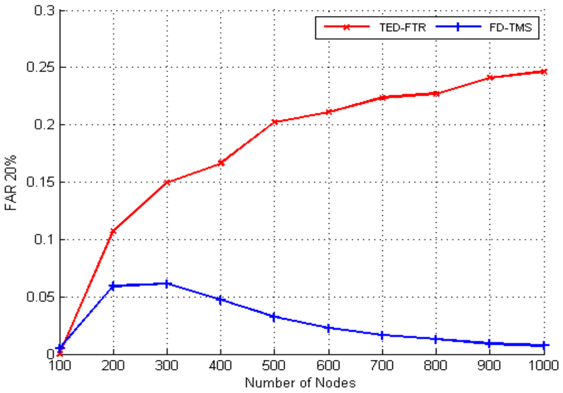


FIGURE 10. FAR for 20% faulty percentage.

because the TMS technique will detect the true event accurately in the high-density network.

2) ENDA

Fig. 11, Fig. 12, Fig. 13, and Fig. 14 presented the ENDA for the proposed FD-TMS method compared with the baseline work TED-FTR method [20]. The analysis employed varied numbers of nodes ranging from 100 to 1000 with a step length of 100 and fault percentages ranging from 5% to 20% with a step length of 5% for a clear and complete evaluation.

The simulation results show that achieving a complete network connection is impossible for networks of 100 and 200 sensor nodes. As a consequence, the accuracy of detection gets lower for both methods. The detection accuracy increases proportionally with the increase in the number of nodes.

Fig. 11 shows that our proposed algorithm achieved more than 20% improvement over the base algorithm with a low percentage of faulty nodes in the network. The FD and TMS strategies enhance the ENDA to reach more than 90% for most cases, especially for high-density networks. The main reason for the high event detection accuracy within our proposed method is related to the developed tenth-man procedures. TMS will verify the majority voting based on the estimated event's location, so most nodes inside the event region will confirm the event occurrence.

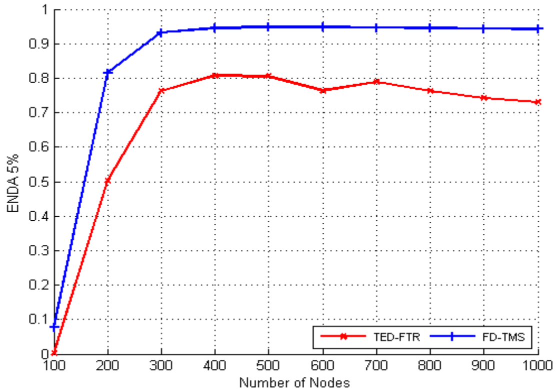


FIGURE 11. **ENDA for 5% faulty percentage.**

For 10% of faulty nodes, Fig. 12 shows that the proposed algorithm reached high stability in its performance. In contrast, the previous method based on the Boyer Moor algorithm presented inadequate performance for event detection because it depends only on the blind majority voting. Allowing all sensor nodes to participate in the voting process makes the previous method inefficient and affects network reliability. In contrast, the proposed method detects the true event no matter how many faulty nodes are around the event region.

Fig. 13 illustrates that our algorithm still presented high ENDA because no faulty nodes with low FD values participated in the voting process. In addition, the TMS plays a vital role by preventing the sensor nodes from obeying the blind majority. TMS allows the sensor nodes to check voters' replies and consider alternative decision-making based on the event's location and the neighbor nodes' geographic locations.

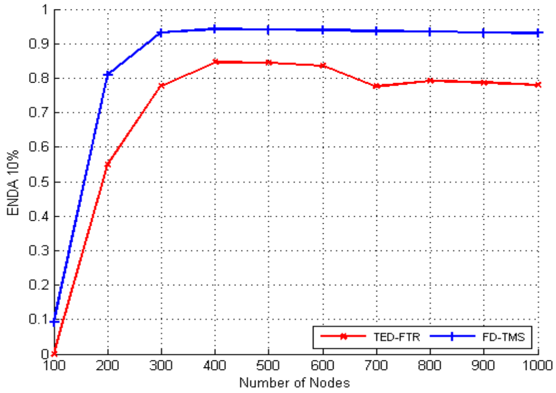


FIGURE 12. **ENDA for 10% faulty percentage.**

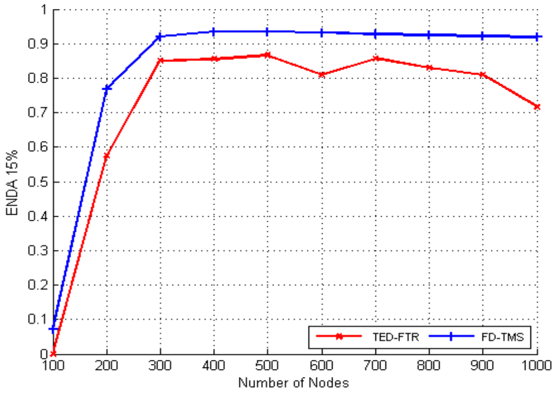


FIGURE 13. **ENDA for 15% faulty percentage.**

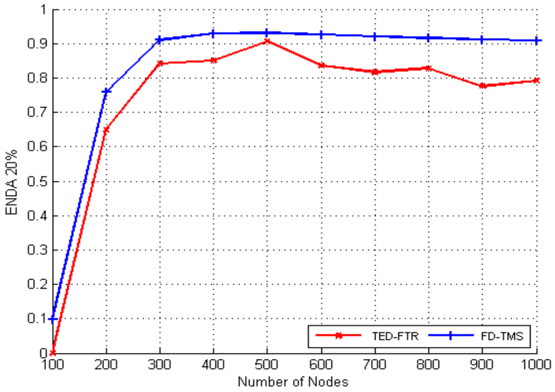


FIGURE 14. **ENDA for 20% faulty percentage.**

Fig. 14 shows that our proposed method still presented high ENDA for 20% of faulty nodes, no matter how the faulty nodes percentage and the network density. The proposed FD_TMS method offers significant improvements compared to the previous method regarding the ENDA. Therefore, the proposed method is able to present a trustworthiness event-driven WSN that meets the required data reliability levels for various applications.

VI. CONCLUSION AND FUTURE WORKS

Sensor nodes in WSNs are not shielded against failures because of their fragility and limited resources, especially when deployed in harsh environments. Differentiating between a true and a faulty reading by the sensor node is a challenging task. Many previous methods dealt with this problem based on only one factor represented by the neighbors' voting. This leads to the collection of inaccurate data and the loss of the network's reliability. This work presented a new method, referred to as Friendship Degree and Tenth Man Strategy (FD_TMS), designed to enhance true event detection in event-driven WSNs.

The first phase in the proposed method is built on the concept of the friendship degree, which has been employed previously in the security fields to detect malicious nodes. Concurrently, the TMS, inspired by military intelligence operations, is introduced for the first time in the context of the WSNs. This strategy is considered the second phase of the proposed method that checks and validates the friendship degree outcomes. The friendship degree will be able to assign FD values for neighbor nodes that indicate the trustworthiness of the nodes. FD will exclude the faulty nodes that are recognized by their low FD value from participating in the voting process. This will lead to more accurate decisions about event occurrence and minimize false alarm rates. Simultaneously, utilizing the MCM to predict the event's location during the second phase, the Tenth-Man strategy will validate the majority voting based on the network topology and the event location. TMS will verify the event decision from the first phase that the FD has done by allowing the opposite decision to be considered. By matching the voters' replies with their location regarding the event's location, TMS will make highly accurate decisions about the occurrence of the true events. Whenever a true event is detected, a report message is transmitted to the BS through a chosen path using geographic routing called TED_FTR. In contrast, the sensor nodes will disregard incorrect readings in a decentralized way via the proposed distributed method without BS interference.

Extensive simulations demonstrate that our suggested solution performs much better than the baseline algorithm based on the Boyer Moor algorithm regarding false alarm rate and event node detection accuracy. The proposed method presented a low false alarm rate with different faulty node percentages within the network. Moreover, the proposed method provided high detection accuracy for the generated events, even with small networks that do not contain a lot of sensor nodes inside the event region.

Future work regarding the proposed method includes enhancing the voting process, especially in terms of energy consumption and latency. The future direction may design a strategy that involves only half the number of neighbor nodes in the voting process. This can be included within the design of a fault-tolerant routing algorithm that can route only the true event reports to the BS in a multi-hop communication style to preserve energy and avoid energy holes inside network topology.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Communication Technology and Network Research Group, Faculty of Computer Science and Information Technology (FSKTM). They also would like to acknowledge the University of Putra Malaysia (UPM) for providing facilities during the preparation of this article.

REFERENCES

[1] M. Ali, B. Salah, and T. Habib, "Utilizing industry 4.0-related technologies and modern techniques for manufacturing customized products—Smart yogurt filling system," *J. Eng. Res.*, Jul. 2023, Art. no. 100144, doi: 10.1016/j.jer.2023.100144.

[2] J. Bhuvana and N. Beemkumar, "Influence of hybrid combination of big data and cyber physical system in case of ind 4.0," in *Proc. 3rd Int. Conf. Advance Comput. Innov. Technol. Eng. (ICACITE)*, Noida, India, May 2023, pp. 2406–2410, doi: 10.1109/ICACITE57410.2023.10182624.

[3] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, and J. C.-W. Lin, "Applications of wireless sensor networks and Internet of Things frameworks in the industry revolution 4.0: A systematic literature review," *Sensors*, vol. 22, no. 6, p. 2087, Mar. 2022, doi: 10.3390/s22062087.

[4] H. Hayat, T. Griffiths, D. Brennan, R. P. Lewis, M. Barclay, C. Weirman, B. Philip, and J. R. Searle, "The state-of-the-art of sensors and environmental monitoring technologies in buildings," *Sensors*, vol. 19, no. 17, p. 3648, Aug. 2019, doi: 10.3390/s19173648.

[5] P. K. R. Maddikunta, S. Hakak, M. Alazab, S. Bhattacharya, T. R. Gadekallu, W. Z. Khan, and Q.-V. Pham, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17608–17619, Aug. 2021, doi: 10.1109/JSEN.2021.3049471.

[6] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Magnetic and radar sensing for multimodal remote health monitoring," *IEEE Sensors J.*, vol. 19, no. 20, pp. 8979–8989, Oct. 2019, doi: 10.1109/JSEN.2018.2872894.

[7] D. Thomas, R. Shankaran, M. Orgun, M. Hitchens, and W. Ni, "Energy-efficient military surveillance: Coverage meets connectivity," *IEEE Sensors J.*, vol. 19, no. 10, pp. 3902–3911, May 2019, doi: 10.1109/JSEN.2019.2894899.

[8] X. Huang, S. Li, and Y. Wu, "LSTM-NV: A combined scheme against selective forwarding attack in event-driven wireless sensor networks under harsh environments," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106441, doi: 10.1016/j.engappai.2023.106441.

[9] Y. Liu and Y. Wu, "Employ DBSCAN and neighbor voting to screen selective forwarding attack under variable environment in event-driven wireless sensor networks," *IEEE Access*, vol. 9, pp. 77090–77105, 2021, doi: 10.1109/ACCESS.2021.3083105.

[10] X. Huang and Y. Wu, "Identify selective forwarding attacks using danger model: Promote the detection accuracy in wireless sensor networks," *IEEE Sensors J.*, vol. 22, no. 10, pp. 9997–10008, May 2022, doi: 10.1109/JSEN.2022.3166601.

[11] S. Gupta, G. Kaur, and P. Chanak, "A deep bi-LSTM based fault detection algorithm for WSNs," in *Proc. IEEE Bombay Sect. Signature Conf. (IBSSC)*, Gwalior, India, Nov. 2021, pp. 1–5, doi: 10.1109/IBSSC53889.2021.9673347.

[12] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian, "Fault tolerance structures in wireless sensor networks (WSNs): Survey, classification, and future directions," *Sensors*, vol. 22, no. 16, p. 6041, Aug. 2022, doi: 10.3390/s22166041.

[13] L. Vihman, M. Kruusmaa, and J. Raik, "Systematic review of fault tolerant techniques in underwater sensor networks," *Sensors*, vol. 21, no. 9, p. 3264, May 2021, doi: 10.3390/s21093264.

[14] G. Kaur and P. Chanak, "An energy aware intelligent fault detection scheme for IoT-enabled WSNs," *IEEE Sensors J.*, vol. 22, no. 5, pp. 4722–4731, Mar. 2022, doi: 10.1109/JSEN.2022.3146853.

[15] L. K. Wardhani, R. A. Febriyanto, and N. Anggraini, "Fault detection in wireless sensor networks data using random under sampling and extra-tree algorithm," in *Proc. 10th Int. Conf. Cyber IT Service Manage. (CITSM)*, Yogyakarta, Indonesia, Sep. 2022, pp. 1–6, doi: 10.1109/CITSM56380.2022.9935888.

[16] N. H. A. Rahman, K. Hasikin, N. A. A. Razak, A. K. Al-Ani, D. J. S. Anni, and P. Mohandas, "Medical device failure predictions through AI-driven analysis of multimodal maintenance records," *IEEE Access*, vol. 11, pp. 93160–93179, 2023, doi: 10.1109/ACCESS.2023.3309671.

[17] R. R. Swain, P. M. Khilar, and S. K. Bhoi, "Underlying and persistence fault diagnosis in wireless sensor networks using majority neighbors co-ordination approach," *Wireless Pers. Commun.*, vol. 111, no. 2, pp. 763–798, Mar. 2020, doi: 10.1007/s11277-019-06884-z.

[18] M. Y. Habash, N. M. A. E. Ayad, and A. E. A. E. Ammar, "Fault tolerant radiation monitoring system using wireless sensor and actor network in a nuclear facility," *Int. J. Electron. Telecommun.*, vol. 67, no. 1, pp. 87–93, Jul. 2021, doi: 10.24425/ijet.2021.135948.

[19] G. Jesus, A. Casimiro, and A. Oliveira, "Using machine learning for dependable outlier detection in environmental monitoring systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 5, no. 3, pp. 1–30, Jul. 2021, doi: 10.1145/3445812.

[20] P. Biswas and T. Samanta, "True event-driven and fault-tolerant routing in wireless sensor network," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 439–461, May 2020, doi: 10.1007/s11277-020-07037-3.

[21] M. Safaei, A. S. Ismail, H. Chizari, M. Driss, W. Boulila, S. Asadi, and M. Safaei, "Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive Bayesian-network-based approach," *Softw., Pract. Exp.*, vol. 50, no. 4, pp. 428–446, Apr. 2020, doi: 10.1002/spe.2785.

[22] M. M. Gharamaleki and S. Babaie, "A new distributed fault detection method for wireless sensor networks," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4883–4890, Dec. 2020, doi: 10.1109/JSYST.2020.2976827.

[23] B. Chander and G. Kumaravelan, "Outlier detection strategies for WSNs: A survey," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5684–5707, Sep. 2022, doi: 10.1016/j.jksuci.2021.02.012.

[24] M. Al Samara, I. Bennis, A. Abouaissa, and P. Lorenz, "An efficient outlier detection and classification clustering-based approach for WSN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6, doi: 10.1109/GLOBECOM46510.2021.9685756.

[25] A. Boussif, M. Ghazel, and J. C. Basilio, "Intermittent fault diagnosability of discrete event systems: An overview of automaton-based approaches," *Discrete Event Dyn. Syst.*, vol. 31, no. 1, pp. 59–102, Mar. 2021, doi: 10.1007/s10626-020-00324-y.

[26] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, vol. 6, pp. 11349–11364, 2018, doi: 10.1109/ACCESS.2018.2794519.

[27] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. J. Jassbi, "Fault management frameworks in wireless sensor networks: A survey," *Comput. Commun.*, vol. 155, pp. 205–226, Apr. 2020, doi: 10.1016/j.comcom.2020.03.011.

[28] J. Calandro, "Employing lesser-known corporate development strategies while avoiding problematic blind spots," *Strategy Leadership*, vol. 50, no. 3, pp. 15–20, Apr. 2022, doi: 10.1108/SL-03-2022-0016.

[29] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Trans. Comput.*, vol. 53, no. 3, pp. 241–250, Mar. 2004, doi: 10.1109/TC.2004.1261832.

[30] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2, Miami, FL, USA, Mar. 2005, pp. 902–913, doi: 10.1109/INFCOM.2005.1498320.

[31] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proc. Workshop Dependability Issues Wireless Ad Hoc Netw. Sensor Netw.*, Los Angeles, CA, USA, Sep. 2006, pp. 65–72, doi: 10.1145/1160972.1160985.

[32] P.-Y. Chen, S. Yang, and J. A. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3832–3842, Jun. 2015, doi: 10.1109/TIE.2014.2350451.

[33] S. J. Bhat and K. V. Santhosh, "Fault tolerant localization based on K-means clustering in wireless sensor networks," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Bengaluru, India, Jul. 2020, pp. 1–5, doi: 10.1109/CONECCT50063.2020.9198415.

[34] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. Jafarali Jassbi, "Novel fault-tolerant clustering-based multipath algorithm (FTCM) for wireless sensor networks," *Telecommun. Syst.*, vol. 74, no. 4, pp. 411–424, Aug. 2020, doi: 10.1007/s11235-020-00663-z.

[35] V. Agarwal, S. Tapaswi, and P. Chanak, "Intelligent fault-tolerance data routing scheme for IoT-enabled WSNs," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16332–16342, Sep. 2022, doi: 10.1109/JIOT.2022.3151501.

[36] R. Prasad and R. K. Baghel, "Self-detection based fault diagnosis for wireless sensor networks," *Ad Hoc Netw.*, vol. 149, Oct. 2023, Art. no. 103245, doi: 10.1016/j.adhoc.2023.103245.

[37] M. Panda and P. M. Khilar, "Distributed Byzantine fault detection technique in wireless sensor networks based on hypothesis testing," *Comput. Electr. Eng.*, vol. 48, pp. 270–285, Nov. 2015, doi: 10.1016/j.compeleceng.2015.06.024.

[38] K. P. Sharma and T. P. Sharma, "RDFD: Reactive distributed fault detection in wireless sensor networks," *Wireless Netw.*, vol. 23, no. 4, pp. 1145–1160, May 2017, doi: 10.1007/s11276-016-1207-1.

[39] L. Alonso and E. M. Reingold, "Analysis of Boyer and Moore's MJRTY algorithm," *Inf. Process. Lett.*, vol. 113, no. 13, pp. 495–497, Jul. 2013, doi: 10.1016/j.ipl.2013.04.005.

[40] R. S. Boyer and J. S. Moore, "MJRTY—A fast majority vote algorithm," in *Automated Reasoning: Essays in Honor of Woody Bledsoe*. Dordrecht, The Netherlands: Springer, 1991, pp. 105–117.

[41] H. Yin, W. Wang, Y. Cao, and H. Zhao, "ENET: Efficient blockchain consensus algorithm based on evil node elimination tree," in *Proc. IEEE 5th Int. Conf. Autom., Electron. Electr. Eng. (AUTEEE)*, Shenyang, China, Nov. 2022, pp. 282–290, doi: 10.1109/AUTEEE56487.2022.9994558.

[42] J. Wang, X. Gao, R. Cao, and Z. Sun, "A multilevel Monte Carlo method for performing time-variant reliability analysis," *IEEE Access*, vol. 9, pp. 31773–31781, 2021, doi: 10.1109/ACCESS.2021.3059663.

[43] A. Barbu and S.-C. Zhu, *Monte Carlo Methods*. Singapore: Springer, 2020, pp. 142–156.

[44] R. R. Swain, P. M. Khilar, and S. K. Bhoi, "Heterogeneous fault diagnosis for wireless sensor networks," *Ad Hoc Netw.*, vol. 69, pp. 15–37, Feb. 2018, doi: 10.1016/j.adhoc.2017.10.012.

GHAIHAB HASSAN ADDAY was born in Basrah, Iraq, in 1981. He received the B.S. degree in computer science and the M.S. degree in routing in wireless networks from the University of Basrah, Basrah, in 2008 and 2012, respectively. He is currently pursuing the Ph.D. degree in wireless sensor networks from University Putra Malaysia (UPM), Selangor, Malaysia.

From 2012 to 2014, he was a Lecturer Assistant with the Computer Science Department, College of Science, University of Basrah. Since 2014, he has been a Lecturer with the Computer Science Department, College of Computer Science and Information Technology, University of Basrah. From 2016 to 2019, he was the Secretary of the College Council, College of Computer Science and Information Technology, University of Basrah. From 2019 to 2020, he was an Assistant Dean for Administrative and Financial Affairs with the College of Computer Science and Information Technology, University of Basrah. His current research interests include routing protocols in wireless sensor networks, fault tolerance framework in wireless networks, trust management in wireless networks, the Internet of Things (IoT), and information security.

SHAMALA K. SUBRAMANIAM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from University Putra Malaysia (UPM), Selangor, Malaysia, in 1996, 1999, and 2002, respectively. She is currently a Professor with the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, UPM. Her current research interests include computer networks, simulation, and modeling.

ZURIATI AHMAD ZUKARNAIN (Member, IEEE) received the B.S. and M.S. degrees in physics and education from University Putra Malaysia (UPM), Selangor, Malaysia, in 1997 and 2000, respectively, and the Ph.D. degree in quantum computing and communication from the University of Bradford, U.K., in 2005.

She has been an Academic Staff with the Faculty of Computer Science and Information Technology, UPM, since 2001. She was the Head of the Department of Communication Technology and Networks, from 2006 to 2011. She was also the Head of the Section of High-Performance Computing, Institute of Mathematical Research, UPM, from 2012 to 2015. She taught several courses for undergraduate students, such as data communication and networks, distributed systems, mobile and wireless, network security, computer architecture, and assembly language. She taught a few courses for postgraduate students, such as the advanced distributed and research method. Her current research interests include computer networks, distributed systems, mobile and wireless, network security, quantum computing, and quantum cryptography. She is a member of the IEEE Computer Society.

NORMALIA SAMIAN (Member, IEEE) received the Ph.D. degree from University Putra Malaysia (UPM), Selangor, Malaysia, in 2017, with a focus on cooperation in wireless multi-hop networks.

She is currently a Senior Lecturer with the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). She is also the Leader of the Wireless, Mobile, and Quantum Computing (WiMoQ) Research Group and also the Head of the Academic Advisor with the Faculty of Computer Science and Information Technology. She is leading a grant project on securing the IoT networks using blockchain technology. Her current research interests include ad hoc network security, cooperation, and trust management in wireless networks, the Internet of Things (IoT), and blockchain technology. She has published several impact factors journals and tier-A conferences related to her fields and has served as a reviewer/technical program committee in international journals/conferences. During the Ph.D. degree, she received the N2 Women Young Researcher Fellowship at IEEE LCN2016 in Dubai.