IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 31, 2021, accepted February 21, 2021, date of publication March 2, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063160

A Method for Fault Tolerant Localization of Heterogeneous Wireless Sensor Networks

SOUMYA J. BHAT AND K. V. SANTHOSH
Centre for Cyber Physical System, Department of Instrumentation and Control Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal 576104, India

Corresponding author: K. V. Santhosh (kv.santhu@gmail.com)

ABSTRACT Location information of nodes in Wireless Sensor Networks (WSN) is essential to identify the origins of events and to act on them. Several localization algorithms are developed for this purpose. In this work, we have considered hop based localization algorithms, which are popularly used in WSN applications. These algorithms use a few reference nodes with location information and localize other nodes with reference to these nodes. But, in practical scenarios, some reference nodes may turn faulty and report incorrect location information to other nodes. This reduces the localization accuracy of the entire network. Therefore, it is essential to identify and filter out faulty reference nodes from the localization process. But, in Heterogeneous Wireless Sensor Networks (HWSN), since both faulty nodes and heterogeneous nodes modify hop distances, it becomes even more challenging to identify only faulty nodes among a set of heterogeneous nodes. In this work, we have reported a fault filtering method that can be used with any of the existing hop based localization algorithms for fault-tolerant localization. This method first normalizes the distance estimations using the communication radius of nodes and then uses the Jenks Natural Breaks algorithm for filtering out the nodes producing inconsistent distance estimations. The reported method is incorporated into existing localization algorithms and tested in 2D/3D, isotropic/anisotropic environments. The results show an improvement of 14%, 52%, and 51% in localization accuracy when tested with DV-Hop, Weighted DV-Hop, and HHO-AM algorithms, respectively.

INDEX TERMS 3D field, anisotropy, fault tolerant, heterogeneous wireless sensor network, Jenks natural breaks algorithm, localization.

I. INTRODUCTION

Wireless Sensor Networks (WSN) are formed using a large number of sensor nodes that communicate with each other through wireless channels [1]. The deployed sensor nodes collect required information about the surrounding events and share the collected data through other nodes to a central location for further processing [2]. Applications of WSN range from military tasks to civilian tasks such as surveillance, intruder detection system, fire detection, oil explorations, and volcanic monitoring systems [3]–[6]. In many of these applications, WSN needs to be deployed in harsh and unreachable fields of interest [7]. For these applications, sensor nodes are usually deployed by random scattering in the required fields of interest using an aerial vehicle. In such cases, the locations of nodes will be unknown. But, location information of nodes is essential for these applications to identify the origins of sensed data and to act on them [8]. Attaching Global Positioning System (GPS) receivers with every node is not a possible option because of increased cost, power consumption, and lower accuracy due to shadowing effects. Hence, several localization algorithms have been developed by researchers to estimate the locations of nodes [9].

The developed localization algorithms use few reference nodes with location information, and locations of unknown nodes are estimated with respect to these reference nodes [10]. Various measurement techniques such as Time of Arrival (TOA), Angle of Arrival (AOA), Received Signal Strength (RSS), connectivity, etc., are used for this purpose [11], [12]. Among the available techniques, connectivity based localization does not need any additional hardware and hence is popularly used for WSN applications [13], [14]. Hop based localization is one such type of method where proximity to reference nodes is measured using the Average Hop

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

Distance (AHD) between any two neighboring nodes [9], [15]. The obtained proximity measures are processed in different ways to get improved localization results.

In reality, it is possible that a few reference nodes do not have correct information about their locations. This may be because of hardware failures, environmental effects, manual errors, or changes in location of a few reference nodes after deployment because of unexpected human or animal movements in those regions. In such situations, they report faulty locations to other nodes, and this results in error propagation [10]. In hop based localization, the faulty locations cause erroneous hop distance estimations affecting the AHD and hence, the localization accuracy of the entire network. Therefore, it is required to identify and filter out the faulty reference nodes causing erroneous hop distance estimations from the localization process for efficient localization. A possible solution is to identify a threshold range for AHD and consider the nodes producing AHD values outside this range as faulty [16]. But, this solution does not work when the network is heterogeneous with sensor nodes of different transmission powers. In HWSN, a node with higher transmission power can communicate to larger distances, whereas a node with lower transmission power will have a smaller communicate range [17], [18]. Hence, the problem becomes more challenging in Heterogeneous Wireless Sensor Networks (HWSN) because the heterogeneity of nodes results in varying hop sizes. Since both faulty nodes and heterogeneity result in non-uniform hop distances, identifying only faulty nodes among the heterogeneous set of nodes is a challenging task.

In this paper, we have reported a fault filtering method that can be used with any of the existing hop-based localization algorithms for accurate localization of a heterogeneous set of sensor nodes in the presence of faulty reference nodes. In this method, the influence of heterogeneity on AHD is overcome by replacing AHD with Average Communication Distance (ACD), which is a function of the communication radius of nodes. This way, errors introduced by the heterogeneity of nodes on the distance estimations are reduced. Next, the nodes causing erroneous distance estimations are identified by checking for inconsistencies using the Jenks Natural Breaks algorithm. The Jenks Natural Breaks algorithm is a standard method for dividing a dataset into a certain number of homogenous classes. The classification is achieved by maximizing the variance between classes and minimizing the variance within classes [19], [20]. The nodes causing inconsistent distance estimations are filtered out from the localization process. The improved distance estimations from the non-faulty reference nodes are then processed by localization algorithms for estimating the positions of unknown nodes. The reported method is tested with various existing localization algorithms in 2D/3D fields, and the results show an improved performance of localization algorithms.

The main contributions of this paper can be summarized as follows. First, an improved distance estimation method based on ACD is reported. This uses the communication radius of nodes to overcome the effect of heterogeneity on distance measurements. Secondly, a faulty node filtering method based on the consistency of ACD is reported. Third, the reported fault filtering method is tested with various hop based localization algorithms, and the obtained results show an improved localization accuracy when tested in 2D and 3D, isotropic and anisotropic fields.

This paper is structured as follows: Section II gives an overview of the related work and the research gaps. Section III contains the problem description and Section IV describes the proposed fault filtering method. Section V contains the analysis of the results and the paper is concluded in Section VI.

II. RELATED WORKS

Several localization techniques have been proposed for WSN which make use of distance measurements based on RSS, TOA, AOA, hop lengths, etc., [11]. Among these, hop based localization techniques have gained the interest of researchers for their simplicity, ease of use, and reduced hardware requirements. Researchers have reported several hop based algorithms aimed at improving different practical problems of WSN. Distance Vector Hop (DV-Hop) [21] is a classic hop based algorithm that uses AHD and the hops between nodes to evaluate the distance among them. The nodes are then localized using the least squares method. Another framework called DV-Hop based genetic algorithm is reported in [22] for situations where few nodes become dead due to depletion of battery with time. This requires the deployment of additional nodes to fill the communication gap. The algorithm only estimates the positions of newly deployed nodes with the help of already localized nodes. Usually, in most cases, localization is a onetime task. But, in scenarios such as landslides, nodes get dislocated within the network. Reference [23] reported a new DV-Hop framework with particle swarm optimization to handle the scenario of node displacement. To improve the positioning accuracy in the case of non-uniform deployment of nodes, an improved DV-Hop localization called MMSDV-Hop is reported in [24]. These algorithms have been developed by assuming homogeneous WSN.

But, in reality, sensor nodes can have different transmission powers forming heterogeneous WSN. Heterogeneity affects the hop lengths and localization accuracies. For localization in HWSN, [25] reported an algorithm based on expected hop progress for HWSN where all nodes' communication ranges are different. Here, an elliptical distance correction method is proposed to calculate the distance between nodes and uses the maximum likelihood estimation method to compute the location information without increasing overhead. Reference [26] reported Harris Hawks Optimization based Area Minimization (HHO-AM) localization algorithm to overcome the effects of heterogeneity and anisotropy. Here, neighbors are classified as incoming and outgoing neighbors, and the positions are estimated using the harris hawks optimization method. Reference [27] reported expected hop progress based analytical algorithm tailored for multi-hop HWSN where nodes have different transmission capabilities.

Most of the developed localization algorithms depend on the accuracy of reference nodes [28], [29]. But, nodes are prone to failures in harsh and unreliable environments. Faulty nodes are likely to report arbitrary readings that do not reflect the true state of environmental phenomena or events under monitoring. Reference [30] reported a localization error detection and correction algorithm for long thin WSN. This algorithm utilizes RSSI and AOA information of radio messages with neighbor nodes for relative estimation of locations. Then, by comparing the difference in location estimation among neighbor nodes against an error threshold, malicious nodes are identified. Reference [31] reported a graph-mining based defect-localization approach called sensor network defect localization. Here, the nodes are sorted according to their suspiciousness from a database of routing trees. Reference [32] reported a super cross-check algorithm that allows location-unknown nodes to successfully detect adversaries within their communication range. Here, after location estimation using a range-based method, a cross-check list containing a series of verification lists is exchanged between neighbor nodes. The nodes sending cross-check list with wrong information are identified as colluder nodes. Reference [33] reported another location discovery method that tolerates faulty reference nodes based on consistency among node signals. This is a range based method that makes use of measurement signals such as RSS to estimate distances. For fault tolerance in hop based localization methods, [16] reported a hop based range-free localization algorithm developed to eliminate the effect of malicious reference nodes. The outliers are removed based on the dynamic threshold obtained via the Pauta criterion.

In the literature, there are only a very few fault-tolerant algorithms that are applicable for hop based localization algorithms. But, these methods have assumed homogeneous network conditions. The algorithms have not been analyzed in heterogeneous networks. In this work, we have reported a fault filtering method for HWSN which can be incorporated with any of the existing hop based methods for better localization accuracies.

III. PROBLEM DESCRIPTION

A WSN of N sensor nodes denoted as $N_1, N_2, \ldots, N_N$ is deployed randomly in the required field of interest. In this network, let L be a small fraction of nodes which are assumed to know their locations. This can be through GPS receivers attached to them or manual placement of nodes at predefined locations. They act as reference nodes. These reference nodes are denoted by $R_1, R_2, \ldots, R_L$ and their locations are represented as (x_{R1}, y_{R1}, z_{R1}), (x_{R2}, y_{R2}, z_{R2}),..., (x_{RL}, y_{RL}, z_{RL}). The other N-L nodes which are denoted as $U_{L+1}, U_{L+2}, \ldots, U_N$ are unaware of their locations. The localization algorithms can be used to estimate the locations of unknown nodes, i.e., (x_{Ui}, y_{Ui}, z_{Ui}), where $i = L+1, L+2, \ldots, N$.

The deployed nodes in the network will have different sensing as well as communication capabilities. Fig. 1 shows an example of a HWSN with nodes of different communication capabilities. Here rc_1, rc_2 and rc_3 are communication radius of nodes N_1, N_2 and N_3.

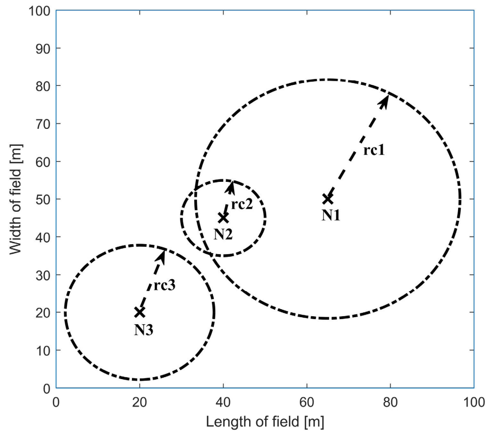

FIGURE 1. Illustration of a HWSN.

After the deployment, localization algorithms are executed for location estimation. These algorithms make use of locations of reference nodes and the proximity of unknown nodes to reference nodes. Proximity to nodes is measured using RSSI, TOA, connectivity, etc., [34]. In hop based localization algorithms, connectivity information among nodes is used to measure the proximity to reference nodes. Here, initially, every node identifies and stores the minimum hops required to communicate with every reference node and the locations of reference nodes in the network through the broadcast mechanism. In the next stage, reference nodes measure the AHD in the network. AHD for a reference node R_i is defined as in (1).

$$AHD_{Ri} = \frac{\sum_{j=1}^{L} d_{R_iR_j}}{\sum_{j=1}^{L} H_{R_iR_j}}, \quad j \neq i \tag{1}$$

where $d_{R_iR_j}$ is the distance between nodes R_i and R_j, and $H_{R_iR_j}$ is the minimum number of hops required by R_i to receive a packet from R_j.

The measured AHD is then used to estimate the distance between nodes which is further utilized by nodes to localize themselves. This method works under the assumption that hop distances are uniform throughout the network. But, in a HWSN, as shown in Fig. 2, the hop distances vary due to the different communication radii of nodes [25]. Assuming uniform hop distance throughout the network will deteriorate the localization accuracy substantially in a HWSN [35].

In addition to this, sometimes it is possible that few reference nodes report their locations incorrectly. This may be because of hardware failures, manual errors or unexpected movement of nodes. Erroneous locations reported by reference nodes affects the localization accuracy of algorithms. For example, consider a reference node N_1 which is at a distance d from reference nodes N_2 and N_3. If both nodes N_2

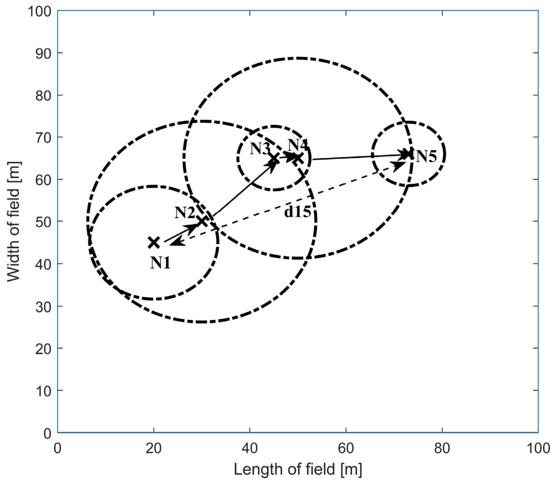

FIGURE 2. Multihop communication in a HWSN.

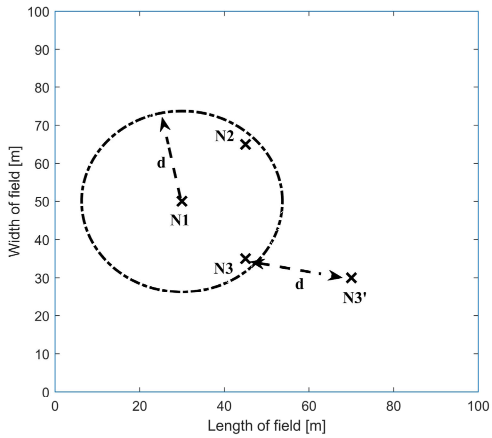

FIGURE 3. WSN with a faulty node.

and N_3 are at hop length *1* from N_1, node N_1 calculates AHD as $(d + d)/(1 + 1) = d$. But, if N_3 reports its location as N_3' which is at distance *2d* from N_1, AHD is calculated wrongly as $(d + 2d)/(1 + 1) = 3d/2$. This is illustrated in Fig. 3.

This error value of AHD is then used by all the location unknown nodes to estimate distances to reference nodes. The error gets propagated to further steps of the localization process. To avoid this, nodes reporting faulty location values need to be filtered out from localization. But, filtering out only faulty nodes in a HWSN where hop distances vary due to both heterogeneity of nodes and faulty nodes is a challenging task.

IV. FAULT FILTERING ALGORITHM

To enhance the localization accuracy in erroneous situations, we have developed a fault filtering method. This algorithm works under the assumption that the maximum number of faulty nodes is less than $L/2$. Here, first, the distance measurement between nodes is improved by measuring the distance as a function of the communication radius of nodes. The obtained distance measurements are then checked for consistency using the Jenks Natural Breaks algorithm [20]. This is an optimal data classification algorithm for one-dimensional values that are not uniformly distributed. The classification is achieved by minimizing the variance within classes and maximizing the variance between classes. The nodes producing inconsistent distance estimations are removed from the localization process. The improved distance estimations from genuine nodes are further used with various localization algorithms. This fault filtering method is found to improve the localization accuracy when used with various hop based localization methods. The details of the algorithm are discussed here.

Step 1 - (Measurement of ACD) : The hop based distance estimations performed using minimum hop counts and AHD suffer greatly in HWSN because of varying hop distances. Also, it becomes difficult to identify the nodes with faulty locations producing inconsistent distance estimations in such environments. In this step, instead of hop distances, communication radius *rc* of nodes are considered. Nodes are assumed to know their communication capability from their transmission power. We have used log normal path loss model to measure the communication radius of nodes. In this algorithm, initially, reference nodes broadcast their location and *rc* to the neighbor nodes. The neighbors will check if the received reference node details are already stored. If it is not stored, the neighbors store the received data, update the received *rc* by adding their *rc* value and broadcast this to their neighbors. If it is already stored, the newly received sum of *rc* information is compared with the saved value and the saved value is updated with the minimum value of the sum of *rc*. This process is repeated till every node in the network has the information on locations of reference nodes and the minimum sum of *rc* required to reach them.

After this, reference nodes in the network measure the ACD as a ratio of the distance between reference nodes and the sum of *rc* between them. ACD for two reference nodes N_1 and N_2 is defined in (2).

$$ACD_{N1N2} = \frac{d_{N1N2}}{rc_{N1N2}} \tag{2}$$

where rc_{N1N2} is the sum of rc stored in node N_1.

This step is illustrated with an example. Consider a WSN with reference nodes $R_1, R_2, \ldots, R_{10}$ and other unknown nodes $U_1, U_2, \ldots, U_{100}$. Let their communication radius be $rc_{R1}, rc_{R2}, \ldots, rc_{R10}$, and $rc_{U1}, rc_{U2}, \ldots, rc_{U100}$ respectively. Locations of reference nodes are (x_{R1}, y_{R1}, z_{R1}), (x_{R2}, y_{R2}, z_{R2}), $\ldots$, $(x_{R10}, y_{R10}, z_{R10})$ which are already known. To estimate the locations of unknown nodes, i.e., (x_{U1}, y_{U1}, z_{U1}), (x_{U2}, y_{U2}, z_{U2}), $\ldots$, $(x_{U100}, y_{U100}, z_{U100})$, their distances to reference nodes need to be measured.

For this, every reference node $R_i, i = 1, \ldots, 10$ broadcasts its location information and communication radius to the neighbor nodes. The neighbor nodes store the received information, add their *rc* values to the received *rc* values, and broadcast them again to their neighbors. This is repeated until

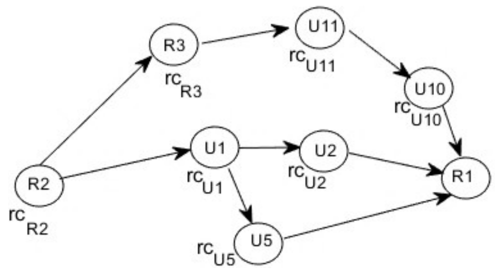

FIGURE 4. **Multipath communication.**

the information on the location of reference nodes and the minimum sum of rc in the path reaches every node.

Suppose $R2$ can send a packet to $R1$ through multiple paths which are $R2-R3-U11-U10-R1$, $R2-U1-U2-R1$, $R2-U1-U5-R1$, as illustrated in Fig. 4. The shortest path from $R2$ to $R1$ is identified by comparing the received rc values. If minimum among $rc_{R2}+rc_{R3}+rc_{U11}+rc_{U10}$, $rc_{R2}+rc_{U1}+rc_{U2}$ and $rc_{R2}+rc_{U1}+rc_{U5}$ is $rc_{R2}+rc_{U1}+rc_{U2}$, then the shortest path from $R2$ to $R1$ is identified as $R2-U1-U2-R1$.

Similarly, the shortest paths from nodes $R3, R4, \ldots, R10$ to $R1$ are found to be,

$$R3-U3-U4-U5-R1$$
$$R4-U10-R1$$
$$\vdots$$
$$R10-U93-U84-U75-R1$$

$R1$ stores the below information.

(x_{R2}, y_{R2}, z_{R2}) and $rc_{R2}+rc_{U1}+rc_{U2}$
(x_{R3}, y_{R3}, z_{R3}) and $rc_{R3}+rc_{U3}+rc_{U4}+rc_{U5}$
.
.
.
$(x_{R10}, y_{R10}, z_{R10})$ and $rc_{R10}+rc_{U93}+rc_{U84}+rc_{U75}$

After this, ACD is calculated by making use of the known locations of reference nodes. ACD for R_1 is calculated as follows.

$$ACD_{R1R2}=\frac{\sqrt{((x_{R1}-x_{R2})^2+(y_{R1}-y_{R2})^2+(z_{R1}-z_{R2})^2)}}{rc_{R2}+rc_{U1}+rc_{U2}}$$
$$ACD_{R1R3}=\frac{\sqrt{((x_{R1}-x_{R3})^2+(y_{R1}-y_{R3})^2+(z_{R1}-z_{R3})^2)}}{rc_{R3}+rc_{U3}+rc_{U4}+rc_{U5}}$$
$$\vdots$$
$$ACD_{R1R10}=\frac{\sqrt{((x_{R1}-x_{R10})^2+(y_{R1}-y_{R10})^2+(z_{R1}-z_{R10})^2)}}{rc_{R10}+rc_{U93}+rc_{U84}+rc_{U75}}$$

Similarly, ACD at every other reference node is measured.

Step 2 - (Consistency Check): In this step, the previously obtained ACD values are analyzed to identify faulty nodes. Since ACD values give a measure of the distance per unit communication radius of nodes, they are not affected by the heterogeneity of the network. The measured ACD values will be uniform at any given node. But, if there exists any node in the network with faulty location information, this data gets propagated to other nodes in the network. In such cases, the measured ACD value for this node will deviate from other ACD values in the network.

From the previous example discussed in Step 1, if node $R2$ reports its location wrongly as $R2'=(x'_{R2}, y'_{R2}, z'_{R2})$, ACD at R_1 is measured as,

$$ACD_{R1R2'}=\frac{\sqrt{((x_{R1}-x_{R2'})^2+(y_{R1}-y_{R2'})^2+(z_{R1}-z_{R2'})^2)}}{rc_{R2}+rc_{U1}+rc_{U2}}$$
$$ACD_{R1R3}=\frac{\sqrt{((x_{R1}-x_{R3})^2+(y_{R1}-y_{R3})^2+(z_{R1}-z_{R3})^2)}}{rc_{R3}+rc_{U3}+rc_{U4}+rc_{U5}}$$
$$\vdots$$
$$ACD_{R1R10}=\frac{\sqrt{((x_{R1}-x_{R10})^2+(y_{R1}-y_{R10})^2+(z_{R1}-z_{R10})^2)}}{rc_{R10}+rc_{U93}+rc_{U84}+rc_{U75}}$$

The measured $ACD_{R1R2'}$ will deviate from the rest of ACD values, $ACD_{R1R3}, ACD_{R1R4}, \ldots, ACD_{R1R10}$.

The inconsistencies among the obtained ACD values are identified using a clustering method called Jenks Natural Breaks algorithm [19]. This algorithm is more suitable for univariate data and finds the best way to categorize the values. This is achieved by searching for the minimum distance between data points and centers of clusters they belong to and maximizing the difference between cluster centers. The algorithm is executed at every reference node. The inputs to the algorithm are the list of measured ACD values and consistency threshold. The consistency threshold is set as a smaller value in ideal network scenarios and in networks affected by noise and irregularities, a larger value is considered. If the difference in the maximum and minimum value of ACD at a reference node exceeds the defined threshold value, the ACD list is considered to be inconsistent. For inconsistent data, the Jenks Natural Breaks algorithm is applied.

In the first part of this algorithm, ACD values are sorted in ascending order and the Sum of Squared Deviations (SSD) for ACD values is calculated. SSD for node R_1 is defined in (3).

$$SSD_{R1}=(ACD_{R1R2}-ACD_{R1mean})^2 + (ACD_{R1R3}-ACD_{R1mean})^2+\ldots \quad (3)$$

where ACD_{R1mean} is the mean of measured ACD values at node R_1.

In the next part, ACD values are classified into two groups with classification index incrementing from 1 to (number of ACD values-1). For classification index i, Group1 is ACD_{R1R2} to ACD_{R1Ri+1} and Group2 is ACD_{R1Ri+2} to ACD_{R1R10}. For every combination of classification, the Sum of squared Class Deviation (SCD) for class means is calculated as in (4).

$$\begin{aligned} SCD_{R1_i} = & (ACD_{R1R2} - ACD_{Gmean1})^2 \\ & + \cdots + (ACD_{R1Ri+1} - ACD_{Gmean1})^2 \\ & + (ACD_{R1Ri+2} - ACD_{Gmean2})^2 \\ & + \cdots + (ACD_{R1R10} - ACD_{Gmean2})^2 \end{aligned} \quad (4)$$

where ACD_{Gmean1} and ACD_{Gmean2} are class means for *Group1* and *Group2*.

Next, the goodness of variance fit (GF) is measured as,

$$GF(i) = (SSD_{R1} - SCD_{R1_i})/SSD_{R1} \quad (5)$$

The value with the highest GF gives the best classification. From the obtained two groups with the highest GF, the group with the highest entries is chosen as the consistent group and the nodes corresponding to these are marked as consistent nodes with a value of 1.

The steps are discussed in Algorithm 1.

Algorithm 1 Clustering of ACD Values Using Jenks Natural Breaks algorithm

Input: List of ACD values, consistency threshold.
Output: Consistent ACD values.
1. **While** (ACD_{max} - ACD_{min}) $>$ **consistency threshold.**
2. **Sort ACD values.**
3. **Jenks Step1:**
4. **Find** ACD_{mean}**, which is mean of ACD**
5. **Find sum of squared deviation** SSD **as shown in** (3)
6. **Jenks Step2:**
7. **For i = 1 to (number of ACD values − 1)**
8. **Classify ACD into two groups ;** $(1 : ACD_i)$ **and** $(ACD_{i+1} : ACD_{end})$
9. **Find class mean for Group1 and Group2,** ACD_{Gmean1} **and** ACD_{Gmean2}
10. **Find** SCD_i **from** (4)
11. **Find** $GF(i)$ **from** (5)
12. **End for**
13. **The value with highest GF is the best classification and** i **is the classification index**
14. **Retain cluster with majority of values**
15. **End While**

The reference nodes broadcast the consistency values and the average of ACD values obtained from consistent reference nodes. After receiving this, every unknown node will add the received consistency values from every reference node. Only the reference nodes with higher consistency values are utilized in the localization process. The distance to these reference nodes is estimated as shown in (6).

$$\tilde{d}_{UxRx} = ACDavg_{Rx} \times rc_{UxRx} \quad (6)$$

where $ACDavg_{Rx}$ is the average of ACD values at node Rx and rc_{UxRx} is the sum of rc values received from reference node Rx at unknown node Ux.

Various localization algorithms can use the obtained distance estimation to consistent reference nodes for the localization of unknown nodes.

V. RESULTS AND ANALYSIS

In this section, the performance of various localization algorithms such as traditional DV-Hop algorithm [21], the recent Weighted DV-Hop [36] and HHO-AM [26] are evaluated with and without fault filtering method under various error scenarios. For this, multiple WSN are simulated by the random deployment of location unknown sensor and reference nodes in various 2D and 3D, isotropic and anisotropic fields as shown in Fig. 5. Various types of deployment fields are chosen to eliminate the impact of field properties on the behavior of the algorithm. The path loss factor in these fields is assumed to be 4 [37]. Sensor nodes are heterogeneous with the transmission powers varying from −5dBm to −15dBm. Tests are conducted at various node densities.

Performance of the proposed algorithm is assessed by Root Mean Square Error (RMSE) [38]. RMSE is defined as,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2 + (\tilde{z}_i - z_i)^2}{n}} \quad (7)$$

where (x_i, y_i, z_i) is the actual node location, $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ is the measured node location and n is the number of localized nodes.

A. INFLUENCE OF FAULTY NODES

Here, the impact of faulty nodes on various localization algorithms and the effectiveness of the fault filtering method in such scenarios is evaluated. A set of 500 sensor nodes and 50 reference nodes are deployed in a square shaped field as shown in Fig. 6. Node density is maintained at $0.01/m^2$ and the reference node ratio is at 10%. The o shaped nodes are location unknown nodes and X shaped nodes are reference nodes. Localization algorithms estimate the locations of unknown nodes by making use of locations of few reference nodes. Sometimes few reference nodes report erroneous locations. In Fig. 6, four reference nodes have reported their locations wrongly. The lines indicate the errors in the reported locations of these nodes.

The nodes are then localized using HHO-AM and HHO-AM with the fault filtering method. Fig. 7 shows the estimated locations of unknown nodes and the error in estimation using HHO-AM and HHO-AM with the fault filtering method. The symbol o indicates the actual locations of sensor nodes and the square symbols indicate the estimated locations. The lines show the error in location estimation. The localization errors are observed to be very high in the HHO-AM algorithm with RMSE of 69.3m as in Fig. 7a. But, by incorporating the fault filtering method to HHO-AM, localization errors were reduced drastically with RMSE of 13.5m, which is shown in Fig. 7b.

Next, the influence of faulty reference nodes on three different hop based algorithms is evaluated here. The algorithms

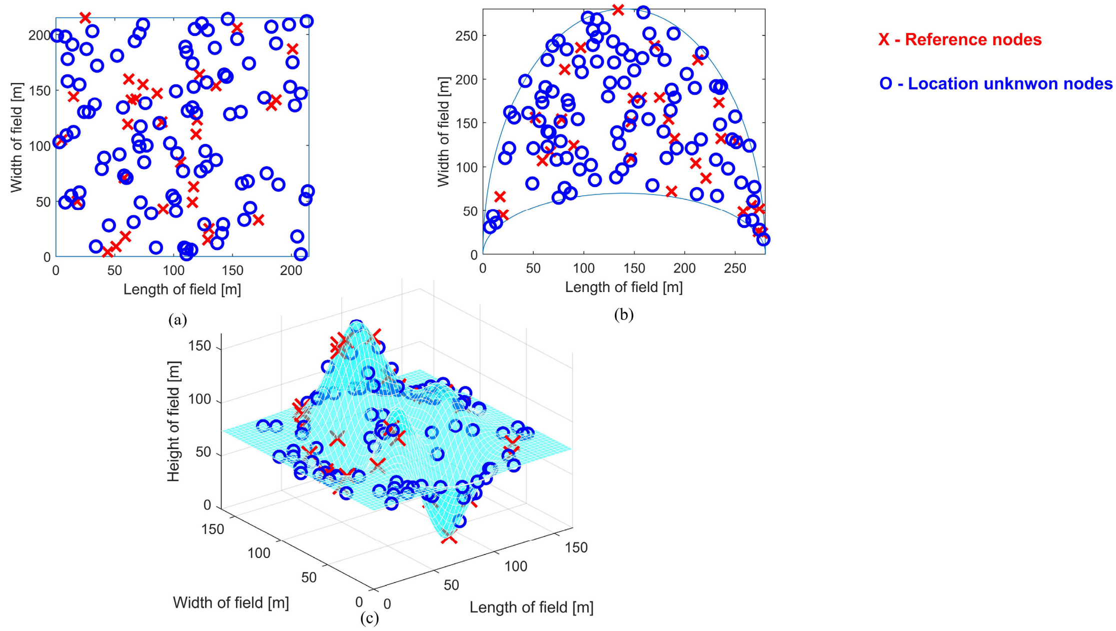

FIGURE 5. Deployment of nodes in (a) square shaped field (b) C shaped field (c) mountain terrain shaped field.

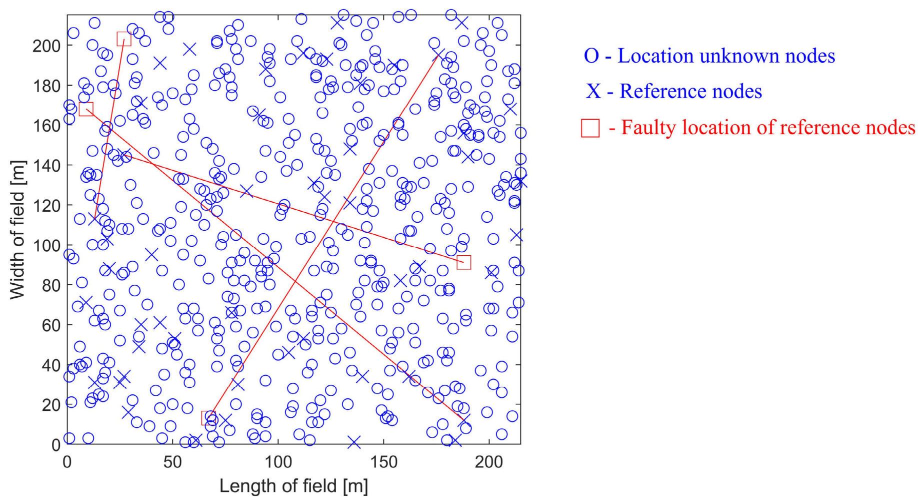

FIGURE 6. Deployment of nodes with errors in locations.

are then updated with the fault filtering method and the localization results are analyzed. The results are shown in Fig. 8.

An increase in faulty node percentage from 0% to 12% reduced the accuracy of localization algorithms drastically in all three fields. Even though Weighted DV-Hop and HHO-AM algorithms performed better than DV-Hop at 0% faulty nodes, their performance deteriorated rapidly with an increase in faulty nodes. But, the fault filtering method was able to filter out faulty nodes effectively in all types of fields and this improved the localization results. An improvement of 14%, 52%, and 51% in localization accuracy was observed with DV-Hop, Weighted DV-Hop, and HHO-AM algorithms respectively.

B. INFLUENCE OF VARYING NODE DENSITY

The reported algorithm is evaluated at varying node and reference node densities. Sensor node densities are varied from $0.01/m^2$ to $0.02/m^2$ in square and C shaped fields. The reference node ratio is set at 10% of node density. Around 10% of the reference nodes are assumed to be reporting faulty locations. Sensor nodes when localized in such scenarios using the DV-Hop algorithm, an average RMSE of 21m was observed. But, after using the fault filtering method with the DV-Hop algorithm average RMSE was reduced to 10m. The fault filtering method resulted in an improvement of 50% RMSE in localization accuracy. This is shown in Fig. 9.

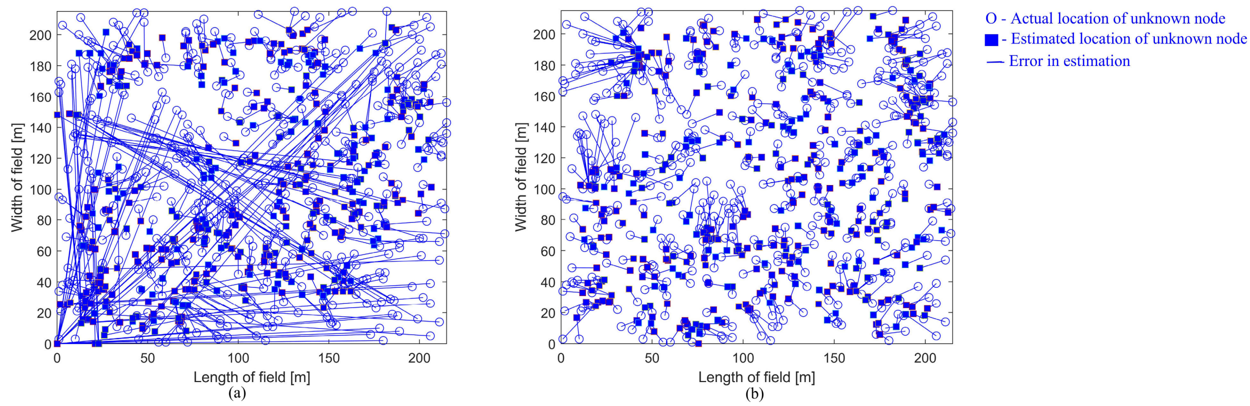

FIGURE 7. Localization errors with (a) HHO-AM (b) HHO-AM with fault filtering.

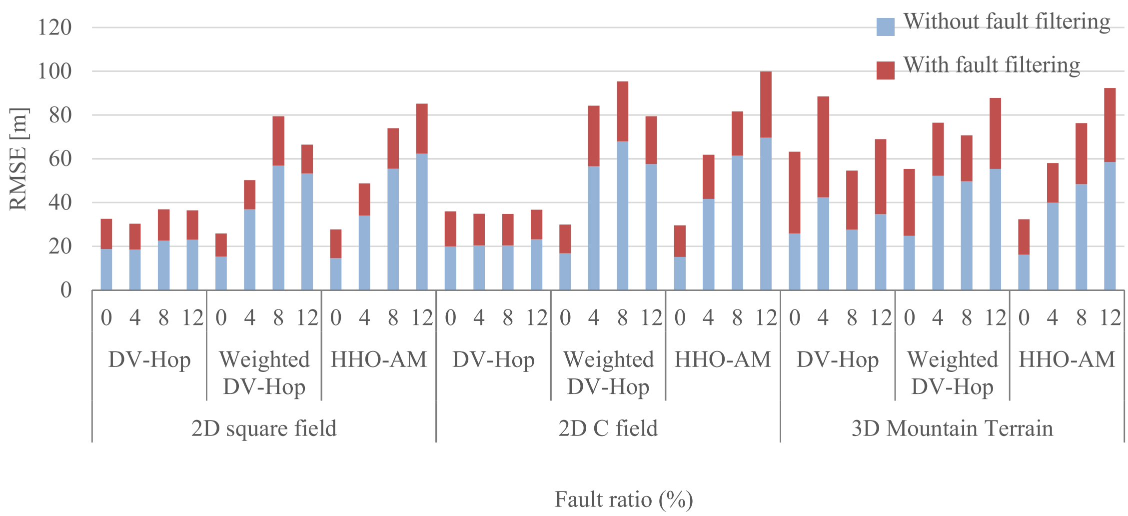

FIGURE 8. Influence of faulty nodes on RMSE.

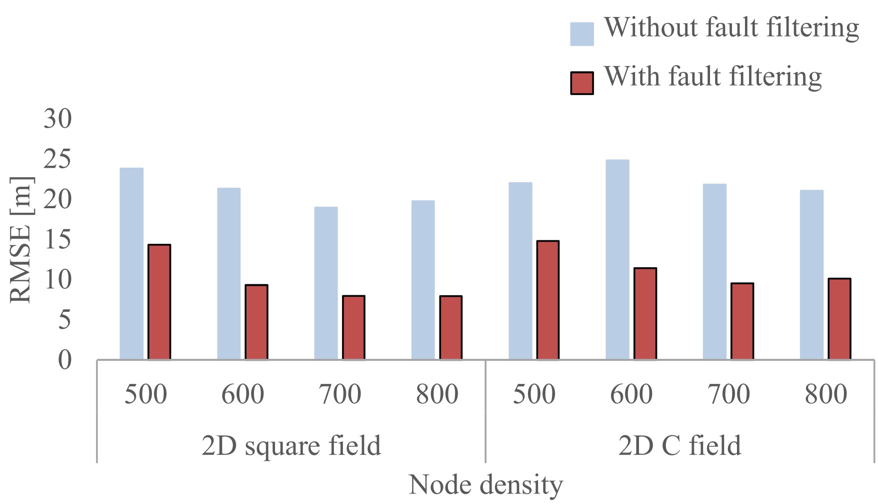

FIGURE 9. Comparison of localization results at varying node densities (a) 2D square field (b) 2D C shaped field.

Next, the reference node density is varied by keeping node density fixed. With the increase in reference nodes, faulty node count also increases. When nodes are localized using DV-Hop, an average RMSE of 21m was observed. The fault filtering method with the DV-Hop algorithm was able to filter the faulty nodes effectively and improve the

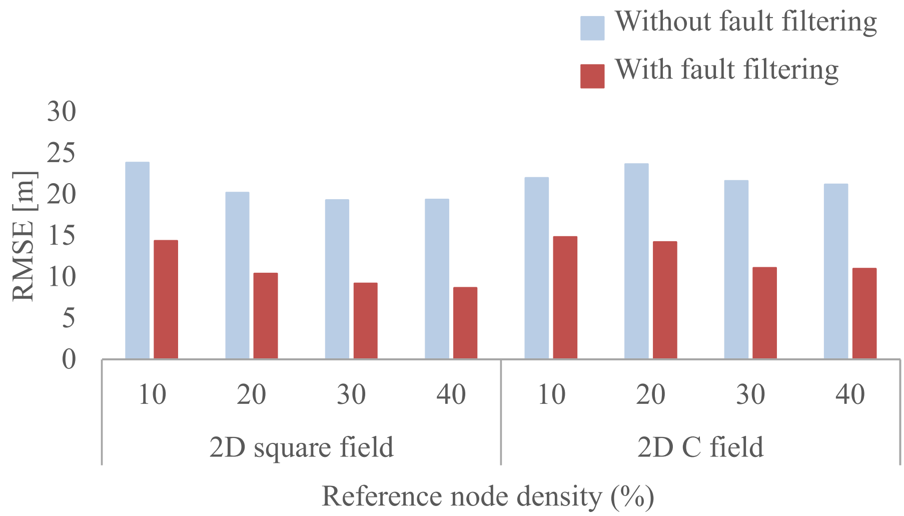

FIGURE 10. Comparison of localization results at varying reference node densities (a) 2D square field (b) 2D C shaped field.

localization accuracy by 45%. The results are illustrated in Fig. 10.

C. ENERGY CONSUMPTION

Energy consumption depends on the communication overhead generated in the network and the computational complexity of the algorithm [39]. The fault filtering method works in two parts. In the first part, reference nodes measure ACD values. The complexity of this part is equivalent to the process of obtaining AHD. Since, in hop based algorithms, the original step of obtaining AHD is replaced by ACD, this doesn't add any additional overhead to the existing algorithms. The second step of the algorithm is to identify inconsistent ACD at the reference nodes using the Jenks natural breaks algorithm. This step makes use of the obtained data from step 1 and hence doesn't add any communication overhead. However, this requires additional computations and the computational complexity of this is $O(k \times L^2)$ where k is the number of classes and L is the number of reference nodes which is a small fraction compared to the overall count of sensor nodes N. Also, since Jenks algorithm is executed only in the reference nodes, additional computation overhead is added only to the reference nodes. If $O(L)$ is the computational complexity of DV-Hop algorithm, by incorporating the fault filtering method, the computational complexity at reference nodes will be $O(L)+O(k \times L^2)$. The computational complexity at sensor nodes will remain to be $O(L)$.

VI. CONCLUSION

Localization algorithms are developed to estimate the locations of randomly distributed nodes in the required fields of interest. These algorithms use few location-aware nodes, and the positions of other nodes are estimated with reference to these nodes. But, in reality, sometimes few nodes can turn faulty and report erroneous location information. This gets propagated and results in inaccurate localization of the entire network. The problem becomes more challenging in HWSN because of varying hop distances. To improve the accuracy of localization algorithms in such scenarios, we have reported a fault filtering method. This method can be integrated with other existing localization algorithms to identify and filter out faulty nodes and make the localization process fault-tolerant. The reported method has been tested with many localization algorithms at various error ratios and node densities in 2D and 3D fields. Simulation results show an improvement of 14% to 52% in RMSE values when tested in error scenarios.

REFERENCES

[1] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla, and S. Jit, ''Coverage and connectivity in WSNs: A survey, research issues and challenges,'' *IEEE Access*, vol. 6, pp. 26971–26992, 2018, doi: 10.1109/ACCESS.2018.2833632.

[2] Z. Sun, Y. Liu, and L. Tao, ''Attack localization task allocation in wireless sensor networks based on multi-objective binary particle swarm optimization,'' *J. Netw. Comput. Appl.*, vol. 112, pp. 29–40, Jun. 2018, doi: 10.1016/j.jnca.2018.03.023.

[3] J. Won and E. Bertino, ''Robust sensor localization against known sensor position attacks,'' *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2954–2967, Dec. 2019, doi: 10.1109/TMC.2018.2883578.

[4] S. Savazzi, U. Spagnolini, L. Goratti, D. Molteni, M. Latva-Aho, and M. Nicoli, ''Ultra-wide band sensor networks in oil and gas explorations,'' *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 150–160, Apr. 2013, doi: 10.1109/MCOM.2013.6495774.

[5] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, ''Deploying a wireless sensor network on an active volcano,'' *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar. 2006, doi: 10.1109/MIC.2006.26.

[6] R. Bhatt, P. Maheshwary, P. Shukla, P. Shukla, M. Shrivastava, and S. Changlani, ''Implementation of fruit fly optimization algorithm (FFOA) to escalate the attacking efficiency of node capture attack in wireless sensor networks (WSN),'' *Comput. Commun.*, vol. 149, pp. 134–145, Jan. 2020, doi: 10.1016/j.comcom.2019.09.007.

[7] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, ''A range-based secure localization algorithm for wireless sensor networks,'' *IEEE Sensors J.*, vol. 19, no. 2, pp. 785–796, Jan. 2019, doi: 10.1109/JSEN.2018.2877306.

[8] S. Abdulaziz AlRoomi, I. Ahmad, and T. Dimitriou, ''Secure localization using hypothesis testing in wireless networks,'' *Ad Hoc Netw.*, vol. 74, pp. 47–56, May 2018, doi: 10.1016/j.adhoc.2018.03.008.

[9] D. Han, Y. Yu, K.-C. Li, and R. F. de Mello, "Enhancing the sensor node localization algorithm based on improved DV-hop and DE algorithms in wireless sensor networks," *Sensors*, vol. 20, no. 2, p. 343, Jan. 2020, doi: 10.3390/s20020343.

[10] A. P. Das and S. M. Thampi, "Fault-resilient localization for underwater sensor networks," *Ad Hoc Netw.*, vol. 55, pp. 132–142, Feb. 2017, doi: 10.1016/j.adhoc.2016.09.003.

[11] S. J. Bhat and K. V. Santhosh, "Is localization of wireless sensor networks in irregular fields a challenge?" *Wireless Pers. Commun.*, vol. 114, no. 3, pp. 2017–2042, Oct. 2020, doi: 10.1007/s11277-020-07460-6.

[12] J. Kumari, P. Kumar, and S. K. Singh, "Localization in three-dimensional wireless sensor networks: A survey," *J. Supercomput.*, vol. 75, no. 8, pp. 5040–5083, Aug. 2019, doi: 10.1007/s11227-019-02781-1.

[13] H. Chen, W. Lou, Z. Wang, J. Wu, Z. Wang, and A. Xia, "Securing DV-hop localization against wormhole attacks in wireless sensor networks," *Pervas. Mobile Comput.*, vol. 16, pp. 22–35, Jan. 2015, doi: 10.1016/j.pmcj.2014.01.007.

[14] L. Gui, F. Xiao, Y. Zhou, F. Shu, and T. Val, "Connectivity based DV-hop localization for Internet of Things," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8949–8958, Aug. 2020, doi: 10.1109/TVT.2020.2998093.

[15] Z. Wang, B. Zhang, X. Wang, X. Jin, and Y. Bai, "Improvements of multihop localization algorithm for wireless sensor networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 365–376, Mar. 2019, doi: 10.1109/JSYST.2018.2851782.

[16] X. Li, L. Yan, W. Pan, and B. Luo, "Secure and robust DV-hop localization based on the vector refinement feedback method for wireless sensor networks," *Comput. J.*, vol. 60, pp. 810–821, Jan. 2017, doi: 10.1093/comjnl/bxx002.

[17] A. El Assaf, S. Zaidi, S. Affes, and N. Kandil, "Efficient node localization in energy-harvesting wireless sensor networks," in *Proc. IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Oct. 2015, pp. 1–7, doi: 10.1109/ICUWB.2015.7324441.

[18] Y. Xing, W. Huangfu, X. Dai, and X. Hu, "Node localization based on multiple radio transmission power levels for wireless sensor networks," in *Proc. Int. Conf. 5G Future Wireless Netw.* Cham, Switzerland: Springer, Apr. 2017, pp. 19–27, doi: 10.1007/978-3-319-72823-0_3.

[19] N. Khamis, T. C. Sin, and G. C. Hock, "Segmentation of residential customer load profile in peninsular Malaysia using Jenks natural breaks," in *Proc. IEEE 7th Int. Conf. Power Energy (PECon)*, Dec. 2018, pp. 128–131, doi: 10.1109/PECON.2018.8684113.

[20] J. B. Leite, J. R. S. Mantovani, T. Dokic, Q. Yan, P.-C. Chen, and M. Kezunovic, "Resiliency assessment in distribution networks using GIS-based predictive risk analytics," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4249–4257, Nov. 2019, doi: 10.1109/TPWRS.2019.2913090.

[21] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommun. Syst.*, vol. 22, pp. 267–280, Jan. 2003, doi: 10.1023/A:1023403323460.

[22] V. Kanwar and A. Kumar, "DV-Hop based localization methods for additionally deployed nodes in wireless sensor network using genetic algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1–19, Mar. 2020, doi: 10.1007/s12652-020-01907-1.

[23] V. Kanwar and A. Kumar, "DV-Hop localization methods for displaced sensor nodes in wireless sensor network using PSO," *Wireless Netw.*, vol. 27, pp. 1–12, Aug. 2020, doi: 10.1007/s11276-020-02446-5.

[24] G. Li, S. Zhao, J. Wu, C. Li, and Y. Liu, "DV-hop localization algorithm based on minimum mean square error in Internet of Things," *Procedia Comput. Sci.*, vol. 147, pp. 458–462, Jan. 2019, doi: 10.1016/j.procs.2019.01.272.

[25] W. Wu, X. Wen, H. Xu, L. Yuan, and Q. Meng, "Efficient range-free localization using elliptical distance correction in heterogeneous wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 1, 2018, Art. no. 1550147718756274, doi: 10.1177/1550147718756274.

[26] S. J. Bhat and S. K. Venkata, "An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107371, doi: 10.1016/j.comnet.2020.107371.

[27] A. E. Assaf, S. Zaidi, S. Affes, and N. Kandil, "Low-cost localization for multihop heterogeneous wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 472–484, Jan. 2016, doi: 10.1109/TWC.2015.2475255.

[28] G. Han, J. Jiang, L. Shu, J. Niu, and H.-C. Chao, "Management and applications of trust in wireless sensor networks: A survey," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 602–617, May 2014, doi: 10.1016/j.jcss.2013.06.014.

[29] X. Ran, L. Shu, M. Mukherjee, Y. Wu, Y. Chen, and Z. Sun, "Impact of irregular radio and faulty nodes on localization in industrial WSNs," in *Proc. Int. Wireless Internet Conf.* Cham, Switzerland: Springer, Dec. 2016, pp. 36–48, doi: 10.1007/978-3-319-72998-5_5.

[30] D. De, "A distributed algorithm for localization error detection-correction, use in in-network faulty reading detection: Applicability in long-thin wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2009, pp. 1–6, doi: 10.1109/WCNC.2009.4917497.

[31] C. Oßner, E. Buchmann, and K. Böhm, "Identifying defective nodes in wireless sensor networks," *Distrib. Parallel Databases*, vol. 34, no. 4, pp. 591–610, Dec. 2016, doi: 10.1007/s10619-015-7189-7.

[32] W. Shi, M. Barbeau, J.-P. Corriveau, J. Garcia-Alfaro, and M. Yao, "Secure localization in the presence of colluders in WSNs," *Sensors*, vol. 17, no. 8, p. 1892, Aug. 2017.

[33] D. Liu, P. Ning, A. Liu, C. Wang, and W. K. Du, "Attack-resistant location estimation in wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 4, pp. 1–39, Jul. 2008, doi: 10.1145/1380564.1380570.

[34] I. Ullah, Y. Liu, X. Su, and P. Kim, "Efficient and accurate target localization in underwater environment," *IEEE Access*, vol. 7, pp. 101415–101426, 2019, doi: 10.1109/ACCESS.2019.2930735.

[35] A. El Assaf, S. Zaidi, S. Affes, and N. Kandil, "Cost-effective and accurate nodes localization in heterogeneous wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 6601–6608, doi: 10.1109/ICC.2015.7249377.

[36] X. Yan, L. Sun, J. Zhou, and A. Song, "DV-hop localisation algorithm based on optimal weighted least square in irregular areas," *Electron. Lett.*, vol. 54, no. 21, pp. 1243–1245, Oct. 2018, doi: 10.1049/el.2018.6512.

[37] C. L. Tseng, F. Y. Liu, C. H. Lin, and C. Y. Lee, "Boundary-improved distance vector-hop localization method with multipower correction for wireless sensor networks," *Sensors Mater.*, vol. 29, no. 6, pp. 675–687, 2017, doi: 10.18494/SAM.2017.1489.

[38] V. Annepu and A. Rajesh, "Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks," *Evol. Intell.*, vol. 12, no. 3, pp. 469–478, Sep. 2019, doi: 10.1007/s12065-019-00239-0.

[39] A. Kaur, P. Kumar, and G. P. Gupta, "A weighted centroid localization algorithm for randomly deployed wireless sensor networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 31, no. 1, pp. 82–91, Jan. 2019, doi: 10.1016/j.jksuci.2017.01.007.

SOUMYA J. BHAT received the Bachelor of Engineering degree from the Manipal Institute of Technology, India, and the Master of Technology degree from the Malnad College of Engineering, India. She has worked as Research and Development Engineer with Nokia Solutions and Networks India Private Ltd. She is currently a Research Scholar with the Department of Instrumentation and Control Engineering, Manipal Institute of Technology. Her research interests include wireless sensor networks, wireless communication, network planning, and localization.

K. V. SANTHOSH received the bachelor's and master's degrees from Visvesvaraya Technological University, Belgaum, in 2004 and 2006, respectively, and the Ph.D. degree from the National Institute of Technology, Silchar, in 2013. He is currently working as an Associate Professor with the Department of Instrumentation and Control Engineering, Manipal Institute of Technology, MAHE. His research interests include sensor fusion, smart sensor, and fault detection and isolation.