A Survey on Holes Problem in Wireless Underground Sensor Networks

PRIYANKA SHARMA¹, RISHI PAL SINGH², MAZIN ABED MOHAMMED³, RACHNA SHAH⁴, AND JAN NEDOMA⁵, (Senior Member, IEEE)

¹Semi-Conductor Laboratory, Department of Space, Mohali 160071, India
²Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar 125001, India
³College of Computer Science and Information Technology, University of Anbar, Ramadi 31001, Iraq
⁴Department of Computer Science and Engineering, Indian Institute of Information Technology, Guwahati 781015, India
⁵Department of Telecommunications, VSB-Technical University of Ostrava, 70800 Ostrava, Czech Republic

Corresponding author: Jan Nedoma (jan.nedoma@vsb.cz)

This work was supported by the Vysoká škola báňská—Technical University of Ostrava (VSB-TUO), Ministry of Education, Youth and Sports in the Czech Republic under Project SP2022/18, Project SP2022/34, Project SP2021/45, and Project SP2021/32.

ABSTRACT The paper aims to put in the forefront a survey on the problem of the existing holes in the Wireless Underground Sensor Networks (WUSNs). Anomalies in wireless sensor networks can weaken the network by impeding its functionalities. These existing anomalies commonly known as holes may cause irreversible devastating problems and may prove fatal to humans. There can be different holes in the network, such as coverage hole, routing hole, jamming hole, sink/ worm hole, etc. This paper intends to discuss various holes problems in WUSNs, reasons behind their occurrence, and different techniques to mitigate the issues. An efficient hole detection technique shall increase network lifetime, use lesser resources and give accurate results with zero false positives. To put a light on the present scenario, methods/ techniques used to cope with different types of holes have been discussed in detail presenting their analysis, constraints, and advantages. Besides, it would also touch upon the comparative analysis in terms of soundness and limitations of these techniques. The study is concluded with future research directions.

INDEX TERMS Connectivity, coverage, holes problem, wireless underground sensor networks.

I. INTRODUCTION

Wireless underground sensor networks (WUSNs) [1]–[6] are an extension of wireless sensor networks in which wireless sensors are placed beneath the earth’s surface. WUSN is an emerging and promising research area. It is a field that will make possible an extensive number of novel applications that were not doable before and shall aid in existing applications in both civilian and military lobbies. The wireless medium acts as a mainstay between sensors for collaborative computing [7]. The same is evident in agriculture monitoring, intruder detection, border patrol, sports ground upkeep, infrastructure monitoring, and environmental monitoring, etc. With shifting research trends toward IoT, WSNs and CRNs (Cognitive Radio Networks) need more attentions. Wireless Sensor Networks (WSNs) comprise small sensor nodes varying from tens to thousands. These tiny nodes are contained with battery power, sensing, communication, and data processing abilities. To put it in another way, WSNs consist of a number of embedded systems that can interact with the environment, process the collected data and pass this information to their neighbor destination nodes.

Owing to the inherent property of energy dissipation, WSNs are prone to failure on the dissipation of the required energy via a subtle number of sensor nodes. It presents challenges, especially in WUSNs as replacing batteries or recharging sensor nodes is a very cumbersome task and not even practically possible in a few working scenarios like deserts, hostile environments, remote areas, etc. Besides energy consumption, other challenges in WUSNs include deploying sensor nodes, node heterogeneity, data agglomeration, underground structure, and fault tolerance. Other than that, in many applications of WSNs [8]–[10], sensor nodes are deployed in an ad hoc manner lacking prudent planning and thorough engineering. Distinguished from traditional wireless networks, WSNs [11]–[13] have denser node deployment, failure-prone/ unreliable nodes, complex computing, power, and memory restraints. Due to various existing
anomalies, sensor nodes are unable to sense/communicate with the other nodes in the network, the issue is known as Holes. There are various factors that contribute to the generation or creation of holes. These factors can be attributed to limited battery leading to failure of a sensor and at times random deployment of sensors resulting in voids as target region remains uncovered. Besides that, holes may also be created due to unintended/ intended malicious nodes in the network. It is required to detect and patch Holes in the network, otherwise, they expand and may impair the whole network [14]. Hence, while dealing with wireless networks, it is pertinent to make a mention that the highest order of seriousness is required. For example, if a particular area remains uncovered, in the case of earthquake prediction and border patrol, the sensor malfunctioning may cost human lives and lead to security breaches, respectively.

A. MOTIVATION BEHIND THE SURVEY

Many challenges are encountered in wireless underground sensor networks owing to several factors like the number of sensor nodes deployed, types of sensor nodes (homogeneous or heterogeneous), deployment scheme, sensing and communication range of nodes, communication links, the environment in which nodes are deployed, and energy consumption by sensor nodes, etc. The holes problem is one of the key issues currently being faced in WSN. Holes can be categorized based on the reason behind their occurrence, life span, and how they affect the network. In any field where sensors are deployed, it is supposed that the sensor network fully covers the target region. Any uncovered region in the area leads to a coverage hole. Coverage holes can occur due to random deployment of sensor nodes at the beginning itself or due to failure of any node or any obstruction at a later stage.

Similarly, if nodes are not able to communicate in the desired manner, it is considered a routing hole. Routing holes can occur due to any failure or greedy forwarding issue. Other categories of holes come under security holes that may occur due to denial-of-service, jamming, or worm/ black/ sink holes. Security holes can be said attack on the network which may adversely affect the network, it can lead to losing information, giving confidential information to a third party, or jamming the communication.

Holes affect the capacity and lifetime of the network and the desired coverage, routing capacity and may present fatal security issues.

All those applications which are meant for uncontrollable/ critical environments and require the random deployment of nodes generally suffer from holes problems. Due to a vast number of applications of sensor networks, hole detection shall be of use for all arenas of sciences and engineering. Our sole focus in this paper is to identify holes in wireless sensor networks deployed beneath the earth. Researchers propose different techniques to cater to the issue, which have their pros and cons. It is required to understand the advantages and limitations of existing techniques to implement existing as well as devise new techniques.

This paper presents a survey on the problem of the holes in the Wireless Underground Sensor Network. Holes in the network have been categorized into different types based on their cause, effect, and life span. Paper details different types of the hole and their identification techniques. It also puts light on the advantages and limitations of these techniques and presents a competitive soundness analysis. Major contributions of the work are as follows.

i. Holes in WUSN, their cause and effects have been discussed. A proper categorization has been done to identify further and analyze the issue.

ii. Holes have been categorized majorly into Coverage Hole, Routing Hole, and Security Hole. There are different types of algorithms to identify these holes. A literature survey has been done and various algorithms have been presented. Researchers have proposed different techniques to identify holes in the network. Techniques are detailed in the study discussing algorithms, pre-requisites, and implementation strategies.

iii. A comparative analysis of hole detection algorithms for different holes has been done, discussing constraints and advantages of every algorithm over others. Such an analysis provides insight into the usefulness of techniques in different scenarios and gives a direction towards devising a new scheme.

iv. A summary of algorithms, tools used, major parameters, computation strategy, layer/ protocol on which it works, sensing model, node deployment strategy has been presented. It shall enlighten researchers about the resource requirements, protocols, working environment, and implementation strategy of existing techniques and work as a baseline for future work.

The rest of this paper is structured as follows. Section 2 presents the problem of the holes in wireless underground sensor networks (WUSNs). In section 3, the classification of the problem of the holes is provided. Section 4, concludes the paper.

II. HOLES PROBLEM IN WIRELESS UNDERGROUND SENSOR NETWORKS

Several incongruities may arise in the wireless underground sensor network and as a result of the same, their functioning can be hampered. A region in the network where sensor nodes cease to work or start behaving maliciously and stop data sensing and/ or communication is known as a hole.

Holes create hindrance in communication and primarily affect the performance of the network. Detection of Holes includes identifying damaged, attacked, or inaccessible nodes. Holes in the network [15] may cause data to be routed along the side of the boundary nodes of a hole which will cause early and untimely energy exhaustion available to a particular node and creation of more holes and enlarge the size of the hole in the network.
Coverage and connectivity [7], [16] are among the significant factors causing holes problems in WSNs, the problem becomes more prevalent when it comes to wireless underground sensor networks. Coverage is a factor that determines how well and up to what extent sensor nodes cover each point in the network.

A wireless sensor network is said to be connected [16] if all nodes in the network are directly or indirectly connected and can communicate with each other. Uncovered/ disconnected areas/regions in the network are known as holes. These holes need to be identified for efficient coverage and effective communication. Holes in wireless networks may create a severe problem.

Due to random deployment of nodes [17], external attacks, and various environmental factors, holes may emerge in the networks and are unavoidable. An area in the sensor network which is not sensed or ceased to be sensed by sensors, and there is no communication in or around due to any temporary or permanent fault, drainage of battery, or any other reason, this area is known as Hole (as shown in Fig. 1). Holes damage sensing and communication functionalities of the network and therefore may create havoc and prove fatal based on their application; thus, their identification and patching are of foremost concern. Timely, hole detection refrain from further energy consumption on the boundary of holes due to congestion. It gives assurance on longevity of network life and also ensures satisfactorily good quality of service. Important factors which need to be taken into account are [3]:

- An aspect such as weather, temperature, soil characteristics to include composition, moisture, and soil homogeneity directly affects the connectivity as well as communication aspect.
- In underground sensor networks, the environment has a direct and significant role in network performance.
- The interment of sensor nodes as well as frequencies of the EM wave also strongly affects underground wireless communication.

III. CLASSIFICATION OF HOLES PROBLEM

Holes are categorized (shown in Fig. 2) based on their time span, effect on the network, and reason of occurring in the network.

**Life Span:** Sometimes a node goes into sleep mode temporarily or there is a temporary failure due to which hole is created at the place, this hole is for that time span called a transient hole, whereas power failure of a node or random distribution of nodes in the network may create a hole which can be only healed by covering the hole region known as a persistent hole.

**Effect on Network:** Holes can have a different effect on the network based on their type. Few holes in the network can hamper the functionality of the network known as functional holes; others may degrade performance and create quality issues known as non-functional holes. Holes that create problems like JAMMING and Denial of Service attacks are security issues or security holes.

**Source of Aberration/Anomaly:** Holes are categorized into coverage holes, routing holes, and security holes based on the cause of the anomaly and the effects they have on the network. Holes in the network can lead to loss of data, premature failure of nodes, as well as the leak of confidential information. Holes’ being a major issue has been addressed by many researchers. These holes, their identification techniques, solutions, and techniques to avoid/heal such holes have been discussed and detailed below to give an insight into the models used by different researchers, compare their advantages and disadvantages over each other. It will hopefully give a direction for developing the new technique to identify holes in the network with good accuracy, fewer resources, and low complexity.

A. COVERAGE HOLE IDENTIFICATION TECHNIQUES

Coverage and Connectivity [16], [18], [19] are two crucial major factors contributing holes problem in the WSNs. If the nodes deployed are far from each other, few areas remain uncovered causing holes. On the other side, if node deployment is denser, sensing accuracy will increase but it comes with a shortcoming of higher collision rate and more energy consumption thereby increasing cost and reduction in life of the network.

The coverage hole problem occurs when some areas remain uncovered in the sensing region [20], [21] due to sparse deployment or node failure. Figure 3 shows the coverage hole problem when sensors use the Unit Disk model using 1-coverage [15]. Various techniques are available (shown in Fig. 4) that are used to identify the coverage holes. These techniques are mainly based on location-based, range-based, and connectivity-based identification.

In location-based techniques, nodes in the network need to know their location. So, special hardware is required for global positioning service. Location-based methods use computational geometry approaches like the Voronoi diagram and Delaunay triangle and give an exact solution. Location-based services like GPS require extra hardware and impose an additional cost, which is not apt for cheaper and tiny sensor nodes.

In range-based techniques, nodes need to know the relative distance between neighbor nodes which is easiest most to determine. Though the technique has lesser complexity and
cost than other techniques, it requires the denser deployment of sensor nodes and distribution must follow a statistical function that is not always practically possible. Most range-based techniques give non-deterministic solutions.

In *connectivity-based techniques*, connectivity information between nodes is needed. Such an approach for finding holes is also known as the topological approach. The approach doesn’t require any location-related information and is very suitable for denser networks. To manage connectivity information, synchronization is needed and flooding of messages is done.

A brief survey of these techniques is given below.

1) **VORONOI BASED TECHNIQUE**

Voronoi diagrams [15], [22], [23] have been used to find out the coverage holes, if any after deployment of all the sensor nodes in the intended/target area. Computation geometry is used to identify holes. Each node makes a Voronoi polygon around it to know if there is any coverage hole or not. It’s required to know the location/position of the neighboring node for every node to make a Voronoi diagram [24]–[27]. The diagram divides the complete region into several Voronoi polygons. In each polygon, there is one such node closer to every point inside the polygon compared to any other node.

To find out the local coverage hole, the area of sensing disk of node contained in the polygon is compared with its corresponding Voronoi Polygon (Fig. 5).

2) **DISTRIBUTED SECTOR COVER SCANNING (DSCS) AND DIRECTIONAL WALK (DW) ALGORITHMS**

Authors have dealt with the coverage holes identification issue in different networks by detecting the sensor nodes located on the boundary of holes. For detecting boundary nodes and coverage holes, two algorithms, Distributed Sector Cover Scanning (DSCS) and Directional Walk (DW) algorithms [28] are used. DSCS works with the scheme that a node’s sensing area is partitioned into a number of sectors by its neighbor nodes, it remodels the boundary nodes identification problem into deducing by checking if all the sensing sectors of a node are in the range of the adjoining neighbors of the sensor node. DW, based on DSCS, detects coverage holes by making a sequence of the boundary nodes which surround the same coverage hole and forming a cluster of the nodes lying on the outer boundary of Wireless Sensor Networks.

In this model, it has been presumed that each sensor node consists of a unique ID representation and is aware of its location; all the nodes can directly communicate with their neighbor nodes which are lying within their communication range and can gather information about their one- and two-hop neighbor nodes. Withal, it is assumed that the sensing and communication range of sensor nodes in the network are regular disks with $R_s$ and $R_c$ as the radii, centered about the node. Few definitions are adopted to further proceed and divide sensing area into sectors, these are: -


\[ a: \text{Absolute angle} \]

As shown in Figure 6, a local Cartesian Coordinate System has been created with node $v_i (x_{V_i}, y_{V_i})$ as the origin, and $v_j$ and $v_l$ belong to the one-hop neighbor list. The coordinate of $v_j$ in local CCS is $(x_{V_j}, y_{V_j})$. The absolute angle refers
to the forward angle from the x-axis positive direction to the destination ray \( v_i \rightarrow v_{\text{Dest}} \) and is denoted by \( \angle \text{Abs}(v_i, v_{\text{Dest}}) \) (where \( \angle \text{Abs}(v_i, v_{\text{Dest}}) \in [0^\circ, 360^\circ] \)).

**b: Central angle**

As shown in Fig. 6, an angle formed by the two rays is called a central angle, denoted as \( \angle \text{Central}(v_{\text{Ray1}}, v_i, v_{\text{Ray2}}) \) (where \( \angle (v_{\text{Ray1}}, v_i, v_{\text{Ray2}}) \in [0, \pi] \)).

Boundary nodes are calculated using DSCS (shown in Fig. 7). Note that sensing area of any node \( S(v_{\text{Node}}) \) can be divided into \( k (k = 1, \ldots, |N_{1,2}(v_{\text{Node}})|) \) sectors by \( N_{1,2}(v_{\text{Node}}) \); each sector is called a sensing sector and the \( k \)th sensing sector is denoted by \( S(v_{\text{Node}}, k) \); if the central angle corresponding to \( S(v_{\text{Node}}, k) \) is \( \angle \text{Central}(v_j, v_{\text{Node}}, v_l) \), then \( S(v_{\text{Node}}, k) \) can also be denoted by \( S(v_j, v_{\text{Node}}, v_l) \). The sectors satisfy the following formula in (1):

\[
S(v_{\text{Node}}) = \bigcup_{k=1}^{\lfloor \frac{N_{1,2}(v_{\text{Node}})}{2} \rfloor} S(v_{\text{Node}}, k)
\]

(1)

After applying DSCS, DW is applied to demonstrate how the Boundary Nodes, BNs that surround the same coverage hole are grouped in a cluster.

Steps are as follow:

1) Set an empty queue \( Q \) to store the nodes clustered in the subsequent steps.
2) Select the node with the least coordinate value, i.e., minimum distance from the list of BNs as the start node \( v_c \), mark it as the current node of DW (denoted by \( v_c \)), and push into \( v_c \). Then \( v_c \) checks from its those 1-hop
neighbors which are included in the set of boundary nodes \(N_1(v_c) \subset \text{BNs}\), whether there are any nodes whose absolute angles belong to \([3\pi/2, 2\pi]\). If there exists any, \(v_c\) selects the node which has the minimum absolute angle as its successor (denoted by \(v_{su}\)), and \(v_{su}\) as the new \(v_c\), is inserted into \(Q\). If not, the scope of the absolute angle is changed to \([0, \pi/2]\) and the procedure listed above is repeated. If \(v_{su}\) is not found in \(N_1(v_c)\), then the same searching method is repeated for \(N_2(v_c)\).

3) Step 2 is repeated until \(v_c\) cannot find its \(v_{su}\) any longer; after that scope of the absolute angle is changed to \([\pi/2, \pi]\) and the same procedure is repeated as in Step 1. Note that the \(v_{su}\) found using this process is named the first inflection node; post this, the searching process is to be continued within the scope \([0, \pi]\).

4) If \(v_{su}\) of \(v_c\) cannot be found in Step 3, change the scope of absolute angle to \([\pi, 3\pi/2]\) and continue the searching method as in Step 3. In this practice, the first found node is marked as the second inflection node, and then the searching process is continued within the scope \([\pi/2, 3\pi/2]\).

5) When \(v_{su}\) of \(v_c\) cannot be found in Step 4, the scope of the absolute angle is changed to \([3\pi/2, 2\pi]\) and the same searching method is repeated as detailed in Step 4. If the first node found in the process is \(v_{st}\) or the one- or two-hop neighbor of \(v_{st}\), then the searching process is complete. Every time the execution of steps from Step 1 to Step 5 is over, a boundary cycle formed by selected nodes is determined. Of all the identified boundary cycles, the first one is the outer boundary of the network, whereas the others are coverage holes.

6) Examine the queue \(Q\); if there is any node in \(Q\) that has its two-hop neighbor as its \(v_{su}\), insert a relay node which is the one-hop neighbor of both the nodes and its \(v_{su}\). The relay node to be chosen should be such that, the sum of the distance between the node and this relay node and the distance between the relay node and \(v_{su}\) is minimum.

Note that the relay node may be a non-BN.

7) Remove all the nodes in \(Q\) and their one-hop neighbors from BNs.

8) If \(\text{BNs} = \emptyset\), repeat the procedure from Steps 1 to 7 until the set of BNs is nil or the number of the nodes left in the set is \(\delta\) (authors have taken \(\delta = 2\)).

3) K-UC AND K-NC PERIMETER BASED TECHNIQUE

Two polynomial-time algorithms k-UC (Unit disk Coverage) and k-NC (Non-unit disk Coverage) [15], [29]–[31] have been given by authors to confirm whether at least \(k\) nodes cover every point in the target region or not where \(k\) is a parameter which indicates the required number of sensor nodes. It is considered that the sensing range of any sensor can be a unit or a non-unit disk. Every deployed node’s location information is the only input required to be provided in the algorithm to estimate the desired multiple coversages. Ink-UC, only neighbor nodes lying within the distance that of two times the sensing range of the node is considered in calculating coverage. Ink-NC, as the sensing range of nodes, is non-uniform, there are different rules for neighbor selection in different scenarios like whether one of the sensor nodes is lying in the sensing range of the other node or both the nodes are in each other’s sensing range. Its perimeter coverage is calculated by a node after detecting the sector of its coverage region which is shared by the neighbor node’s sensing range. By looking at perimeter coverage, the node figures out if the whole perimeter \(2\pi\) is covered by existing neighbor nodes.
with the desired degree of coverage or not. A central controller entity is recommended by authors to detect coverage holes; an entity that can gather the information about the regions/areas which are not covered to the required degree from each such node. So that, the required number of nodes can be placed to get the desired coverage after detection of network voids. However, as it is a centralized approach, the approach comes with scalability constraints.

4) BLIND SWARM HOMOLOGY BASED TECHNIQUE
A coverage determination method containing blind swarm [32] of robots has been proposed which uses only distance estimation and requires no localization [33]. The method can be used for every planar region of any shape and size. Following are assumptions that state communication capabilities and take care that domain is not too haggard or pinched:
- Each node consists of a unique ID and is capable of hearing its neighbors as weak (larger radius \(r_w\)) or strong signal (smaller radius \(r_s\), depending on how far is that neighbor node.
- Nodes have symmetric sensing/broadcasting radius \(r_c\) called covering radius.
- Covering radius \(r_c\) and communication radius \(r_s, r_w\) satisfies conditions:
  \[ r_s \leq \sqrt{3} r_c \quad \text{and} \quad r_s \leq \sqrt{\frac{3}{13}} r_w \]
- Nodes lying within a fence radius \(r_f\) can detect network boundaries.
- Domain D-C is connected where C represents:
  \[ C = \{ x \in D : \|x - \partial D\| \leq \frac{1}{2} r_s + r_f \} \]
- Any curve within fencing radius of boundary should have injectivity radius \(\frac{1}{2} r_s\) and external injectivity radius should be a minimum of \(r_s\) and \(r_f + \frac{1}{2} r_s\).

Various graphs have been built using simplicial complexes. These graphs are communication graph \(G_c\) for strong signals, communication graph \(G_w\) for weak signals, fence subgraph \(F_c\) for strong signals, and fence subgraph \(F_w\) for weak signals. Fence subgraphs are maximal subgraphs of communication graphs for those nodes which lie in the fence radius of the network boundary. These graphs are used to generate simplicial complexes. An algebraic variant of simplicial complexes, homology [31] is used to determine coverage and holes in the network. Homology coefficient [32], [33] has been used to avoid any round-off error.

5) TWO ANCHOR (TA) AND CYCLIC SEGMENT SEQUENCE (CSS) BASED TECHNIQUE
Authors have proposed two distributed k-coverage verification and hole detection schemes Two Anchor (TA) and Cyclic Segment Sequence (CSS) [37] without using location information. Most hole detection algorithms require location information of sensor nodes, which adds an extra burden on sensor nodes as it requires additional hardware and software capabilities. These schemes are deterministic and give assurance on detection results. In the network, the sensing radius of every sensor node has upper bound SR, and the transmission radius has lower bound TR. Nodes are unaware of their locations but they can assess their distance from neighbor nodes [38], [39]. As every node has distance information in its local vicinity, therefore coverage of a node is preserved locally independent of the coordinate system. So, a local polar coordinate system has been formed to calculate the coordinates of any node. First, a scheme has been proposed to verify 1-coverage and after that, it has been extended to verify k-coverage for \(k \geq 1\).

In a two-anchor scheme, every node creates a local polar-coordinate system by selecting two of its neighbor nodes termed anchors as reference points. The location of other neighbor nodes is calculated with reference to this local coordinate system. When all neighbor coordinates are marked, then intersection points of the node’s sensing boundary [40], [41] with sensing boundaries of neighbor nodes are calculated. It is verified if every intersection point is covered or not [29]. The scheme is robust to errors in measurement of the distance between nodes but is applicable only in cases when \(TR \geq 4 SR\).

A cyclic segment sequence scheme is a localized method for hole detection. It works on the scheme that for any sensor node \(x\), a segment of the sensing boundary of node \(x\) is covered by the sensing range of the neighbor sensor node. As per the scheme, there will be a cyclic sequence of such segments that sensing range boundary of another sensor node if the sensor node doesn’t fall on the boundary/border of the coverage hole. It is shown that the scheme can be directly used to verify 1-coverage without having any need for the coordinate calculation of neighbor nodes. To extend the solution to verify k-coverage for any \(k \geq 1\), a local polar coordinate system has been defined using the segment sequence. Coordinates of every neighbor node and the local polar coordinate system are used for k-coverage verification. CSS has relaxed transmission radius requirements in comparison to the TA scheme. It can be used where \(TR \geq 2 SR\) but CSS has more sensitivity toward errors occurring in measuring distance.

6) UNMANNED AERIAL VEHICLE (UAV) BASED TECHNIQUE
An Unmanned Aerial Vehicle (UAV) [15], [17], [42] deals with the issue of network deployment and ensures proper connectivity. The network can be deployed by the flying robot, named AVATAR [43] based on pre-computed network topology. It can determine connectivity gaps in the deployed network and generate a plan to repair the network and ensure complete connectivity. The sensors, after getting deployed, figure out the connectivity amid nodes and transmit the same to the airborne UAV.

The deployed network’s connectivity map is computed and analogized with the required topology and the disconnections in the network are recognized. Deployment points inside the disconnected regions are evaluated to fix up the issues and refurbish the network. After the computation, UAV is set to
again deploy new sensor nodes at the points where deployment is needed. Deployment of additional sensors fixes up failing sensor nodes in the network (achieving multiple coverage and connectivity) and makes it denser.

The process is implemented in three phases:-

1) In the first phase, initial network deployment is put into practice. Control algorithms are developed which give direction to the flying robot during the deployment of sensor network with stated communication topology. Having a predefined deployment scale and network topology, topology is embedded in the plane, and locations of nodes are extracted from ensuing embedding. Network topology is depicted in a graphical form where nodes in the graph represent sensors in the network and edges represent the connectivity among various sensor nodes. The nodes which are within communication range of each other showing bidirectional communication are connected by edges in the graph. Computing actual GPS coordinates are not required for deploying sensor nodes after embedding such topology in the plane. The embedding scale depends on the communication range among different nodes. After that, the resulting points are sorted and converted to the waypoints such that the required trajectory of the robot for covering the points is optimized.

2) Deployed network’s connection topology is measured, and a comparison is made with the desired topology in the second phase. If topologies match, the process is complete, if not, the 3rd step is executed.

3) A Connectivity repair algorithm is used to repair the network. The connectivity algorithm generates a connectivity graph using which new deployment locations are computed and the network is repaired generating the desired topology. Measured connectivity graph is compared with the desired topology using graph isomorphism algorithms.

Any kind of potential failure in sensor nodes of the network can be detected as well as persistence in connectivity can be assured by running the last two phases at any given point in time. Sensor density in the region can also be increased using the same approach.

7) ČECH COMPLEX AND RIPS COMPLEX BASED TECHNIQUE

Čech complex and Rips complex [45]–[47] have been adopted by authors to find out coverage holes. A comparative analysis of Čech complex and Rips complex [44] has been done in respect of coverage holes by varying ratio of communication radii ($R_c$) to sensing radii ($R_s$) of a sensor node. It has been observed that if this ratio, $R_c/R_s$ is two or more than two and condition favoring that if there is any hole found in Rips complex, it will also be there in Čech complex too. Coverage holes are categorized as non-triangular or triangular based on the results.

**Definition 1 (Čech Complex):** Taking a collection of sets $u = \{U_a\}$, Čech complex of $u$, $\hat{C}(u)$, is the abstract simplicial complex for which $\text{k}$-simplices conform to a non-empty intersection of $\text{k} + 1$ distinct elements of $u$. $u$ is sensor cover, it is the union of a set of points $X$ of discs having radius $r_c$.

According to Čech Theorem, for suitably good sets $u_a$, the Čech complex $\hat{C}$ and the union $U_a U_a$ possess the same topology.

**Definition 2 (Rips Complex):** Having a set of points $X = \{x_u\} R^m$ in Euclidean n-space and having a fixed radius $\epsilon$, the Rips complex of $X$, $R(X)$, is defined as the abstract simplicial complex for which $\text{k}$-simplices equate to unordered $(\text{k} + 1)$-tuples of points in $X$ which in pair lies within Euclidean distance $\epsilon$ of each other. The Rips complex has been considered perfect for communication networks because the whole complex is figured out by pair-wise communication of data.

Network connectivity is denoted by Connectivity Graph $G(V, E)$ [44], here $V$ depicts the set of sensor nodes, and $E$, the set of edges connecting these sensor nodes. It is presumed that,

1) The sensor nodes which are within Euclidean distance $R_c$ have an edge connecting them.
2) Target field has internal sensors as well as sensor nodes located on the boundaries known as fence sensors. Each fence sensor node is having two fence neighbor nodes.
3) Though sensor nodes don’t know their location, each and every node can predict/know by using models as shown in [10] or as presented in [15] that if it is an internal node or fence node. Indeed, it is a straight supposition chosen by a good number of existing connectivity methods [14], [15] or range-based methods [10], [11].
4) The network is connected.

Čech complex and Rips complex have been opted for characterizing coverage holes.

Čech complex and Rips complex possess relationship as below:

$$R_{R_c}(V) \subset \hat{C}_{R_c}(V) \subset R_{2R_c}(V), \quad \text{if } R_c \leq \sqrt{3} R_s$$

Using the above relationship a set of corollaries can be defined. Consider, $\lambda = R_c/R_s$

**Corollary 1:** When $\lambda \leq \sqrt{3}$, if there is no hole found in Rips complex $R_{R_c}(V)$, there must not be any hole in Čech complex $CRs(V)$.

**Corollary 2:** When $\lambda \geq 2$, if there is no hole found in Rips complex $R_{R_c}(V)$, there must not be any hole in Čech complex $CRs(V)$.

**Corollary 3:** When $\sqrt{3} < \lambda < 2$, there is no assurance of relation in between Rips complex $R_{R_c}(V)$ and Čech complex $CRs(V)$ in terms of holes.

In Figure 8, there are three coverage holes in the network; hole1 is enveloped by nodes 1, 2, 3, hole2 by 2, 4, 5, and hole3 by 2, 3, 5, 6 respectively. It can be seen that all of the holes got detected in the Čech complex, whereas in the Rips complex, only the hole bounded by nodes 2, 3, 5, 6 is detected. Verily, as demonstrated in [32], in the Čech complex any coverage hole present in the sensor network can be detected.
The algorithm finds boundaries/ borders of the holes which have been detected/ found by the Rips complex [44]. Following steps are followed in the process.

1) Information about 1- and 2-hop neighbors is collected by every sensor node and a neighbor graph is generated using the information. This is achieved by sending broadcast messages to all the nodes.

2) Using the neighbor graph, its own status whether it is a hole boundary or not is checked by every node. If a Hamilton cycle is formed in the neighbor graph, then the node is a non-hole boundary node, if not then it is a hole boundary. Nodes have been divided into two types; internal and fence nodes. Fence nodes are those which are on network boundaries. For each internal node, the node itself is not added in the neighbor to check the Hamilton cycle whereas, in the case of fence nodes, the generated neighbor graph also contains the node itself to check if the Hamilton cycle is there in the new graph or not?

3) Using step (2), when each node has found its status whether it is a hole boundary or not. The same is broadcasted by the node to its neighbor; hence each node will know the type of its neighboring nodes. After that, a search for cycles bounding holes is initiated by some of the nodes. As it is a distributed algorithm, so no central entity is available to decide which nodes shall start the process. It is willy-nilly presumed that any hole boundary node which has the minimum id amongst all of its hole boundary neighbor nodes or doesn’t have any hole boundary neighbor, the node is chosen by itself to begin the procedure/ activity.

4) There are possibilities that some cycles detected using the procedure followed in step 3 are not minimal or more than one cycle are bounding the same hole. So minimization is required to choose optimal ones amid all detected cycles. This is realized by each node in the cycle in a distributive manner.

B. ROUTING HOLE IDENTIFICATION TECHNIQUES

A routing hole [15], [48]–[50] is an area in the network where either there are no nodes present or the nodes available in the area are not able to take part in the actual data routing owing to certain reasons. Voids during sensor node deployment or sensor node failure can be the possible reason for these holes. Sensor nodes fail due to a number of different reasons which may include battery depletion, node malfunctioning, or any other external event which has the potential to physically been categorized into three types: represented by 0, 1, and 2, here, if the last node in node sequence is a hole boundary node, then it is denoted as Type 0; if the last node is a non-hole boundary node, then it is denoted by 1 and if the last two visited nodes are non-hole boundary nodes then the message is denoted as Type 2. Some hole boundary nodes may not be discovered, if those nodes continue broadcasting then it is also possible that a few holes remain undetected. But if all the nodes in the network broadcast the message (whether these nodes are hole boundary or not), the number of messages will increase manifold and may overwhelm the network. To make a trade-off between two, a probability p is set for non-hole boundary nodes. If a type 0 message is received by a non-hole boundary node, the message type is set to 1 by the node and broadcasted with probability (1 − p). If a type 1 message is received by a non-hole boundary node, the message type is set 2 by the node and broadcasted with probability (1 − 2p). When a type 2 message is received by a non-hole boundary node, it will be discarded by the node. If any type of message is received by the hole boundary node the message type is set to 0 by the node and will be broadcasted.

In addition to this, any message received by non-hole boundary fence nodes will not be forwarded. Every Node will attach its ID in the ‘Node_ID’ field in sequence while broadcasting the message. Using this scheme, the message contains IDs of all the nodes from the beginning to the current one in sequence. Each Node needs to verify by checking all the messages it has received if there exists in any cycle. In any two messages, checking Node_ID field, if the first IDs are found matching and the last IDs are unlike and nodes are not neighbors, then it is considered that a cycle is there. The cycle is constructed by the node and kept. The cycle created may not be minimal, which can be found by the next step.
TABLE 2. Models for coverage hole detection – advantages and constraints.

| Researcher          | Model/ Concept Used                  | Constraints                                                                 | Advantages                                                                 |
|---------------------|--------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Wang et al. [22]    | Voronoi Diagram                      | All Nodes needs to know their own as well as their respective neighbour’s location. | Less communication complexity.                                           |
| Zhao et al. [28]    | DSCS and DW Algorithm                | All nodes need to know their location. The sensing range is considered the same as the communication range of nodes which leads to the denser placement of nodes without need. | DSCS shows a very good performance in detecting boundary nodes and DW works well in identifying coverage holes of different sizes and shapes. The algorithms effectively heal coverage holes, improve data forwarding efficiency, and prolongs the lifetime of WSN. |
| Huang et al. [29]   | k-UC and k-NC polynomial-time algorithm | Location information is needed. To detect coverage holes, a central controller entity is recommended by authors to gather the details of separate segments or regions which don’t have the required degree of coverage and thus discover and patch existing holes. This will lead to single point failure as well as have scalability issues. | The algorithm provides a solution based on perimeter coverage. It supports unit as well as non-unit sensing disk sensors. |
| Silva et al. [32]   | Blind Swarm                          | Sensor nodes hardware may not be compatible with bimodal signal – weak and strong, reading. Centralized computation puts extra load and may lead to performance deterioration. | The algorithm requires very little information about the environment to detect holes and determine coverage. A rough estimation of the distance between nodes is used to compute coverage. It works for the planar region of any size and shape. |
| Bejerano et al. [37] | Two Anchor (TA) and Cyclic Segment Sequence (CSS) | TA scheme requires that transmission range should be at least four times sensing range which is a major drawback where as CSS is very sensitive toward measurement error. | Algorithms that are coordinate-free require only distance information and provide a solution to verify k-coverage. |
| Corke et al. [17]   | Unmanned Aerial Vehicle (UAV)        | It needs predetermined topology information. The experimental results in the network deployment phase demonstrate that desired network topology is difficult to attain using aerial deployment. | Flying robot repairs the network. These types of networks are preferred for different activities like wide-ranging environmental monitoring and for issuing a command and controlling critical scenarios. |
| Yan et al. [44]     | Čech complex and Rips complex        | Holes have been divided into Triangular and Non-triangular ones. Models use connectivity information. When the communication range (Rc) and sensing range (Rs) are such that Rc < Rs, in such a scenario, only connectivity information is not sufficient to discover triangular holes. Therefore, holes may remain un-discovered sometimes. | Čech complex and Rips complex have been used to identify holes. A connectivity-based distributed algorithm has been used to discover non-triangular holes. |

destroy the nodes. Sometimes, routing holes may get created due to local minimum phenomenon which is frequently seen event in geographic greedy forwarding; a hole that does not exist at that point in time. In Fig. 9, node x wants to send data to the destination node. It searches in its 1-hop neighbors for the node which is geographically nearer (greedy forwarding) to the destination than the node itself for forwarding the traffic. The node x is unable to locate any such 1-hop neighbor node which is nearer to the destination node than the sending node itself. Hence, the process stops as there are only routes that require nodes to move the packet afar from the destination to node b or yad-interim which will not be used by x due to greedy forwarding. This is an example of local minima and normally can be seen when there is a routing hole.

Multi hop routing protocols are divided into three categories (shown in Fig. 10) based on the number of copies of a single data packet that are being forwarded to the destination at the same time; can be categorized as single-path routing, multi-path routing, and routing with greedy forwarding as geographic-based routing. In single-path routing, only one copy of any data packet is transmitted via a single path in the network. Whereas in multi-path routing, multiple numbers of copies of every single packet are forwarded in parallel via different routes to the destination.

A brief survey of these techniques is given below.
1) DIRECTED DIFFUSION (DD) SINGLE PATH ROUTING ALGORITHM

It is a data-centric communication protocol designed especially for wireless sensor networks i.e., all communication is on named data [15], [51]–[53]. Data is represented in form of attribute-value pairs [54]. An interest for named data is generated by the sink node which is circulated in the whole network. That interest is received by nodes by establishing a gradient towards neighboring nodes. If a node is able to produce the desired data mentioned in the interest, it starts sending exploratory data at a rate stated in the interest message to all its neighbor nodes from which interest for the data has been received. The node (sink) which initiated the interest receives this exploratory data from multipath. Data received can be cached, or transformed by the intermediate nodes as well as on the basis of previously cached data, node can direct interests. One particular neighbor is chosen by sink for receiving data via sending a unicast positive reinforcement message to the sender node. Now, the sender node, which gets the positive reinforcement message, will transfer data along the reinforced gradient at higher rates to the sink node. Directed Diffusion can be categorized as single-path or multi-path routing based on number of paths reinforced by sink node. It can also function as a multipath routing protocol if other alternate data sending routes are available as well. Therefore, the reinforcement messages are not required in a one-phase pull; the lowest latency path is reinforced implicitly. There are two disadvantages of one-phase pull over two-phase pull. First, in OPP it is assumed that communication between nodes is symmetric. Second, it is required that interest messages should have a flow-ID. Though creating flow-ID is quite simple, but the size of interest will grow with the number of sink nodes by the inclusion of flow-ID. By comparison, though, the interest messages being generated an increase in number with a two-phase pull in ratio to the number of sink nodes, so here, the cost is reduced.

2) PIVOT-INITIATED PATH REPAIRING SINGLE PATH ROUTING ALGORITHM

In this approach, before data transmission, each sink node initiates a process to set the global lowest-cost path and establishes an optimal path from each sensor node to the sink node. Here, energy consumption, remaining energy, delay, hop number, etc., alone or in combination may represent the cost value. Whenever a packet is to be sent to the sink, the optimal path is calculated using IDs recorded in the source node’s routing tables as well as in its successor nodes and followed to transfer the data packet from the source node to the destination on its optimal path. During data delivery, transmission can be broken due to any kind of channel error or node failure on the route. The node which is forwarding a data packet is accountable for guaranteed delivery of the message and the conformance that the node next to it in the routing path has received the data packet successfully. The transmitting node may implement this by monitoring the packet through acknowledgment receiving in a time period $T_f$. The transmitting node keeps the message in its buffer till it doesn’t get any acknowledgment/receipt. It may be
possible that acknowledgment is not received due to node or channel failure. Any temporary breakage in the path because of interference, obstacle, or collision in the channel is known as Channel error whereas a permanent breakage of the path due to energy depletion, physical damage, or malfunctioning of sensor nodes is known as node failure. When failed acknowledgment receipt is detected by the transmitter, it is assumed that its downstream node is “dead” and search for an alternative path is started to avoid the “dead” node in its neighborhood.

The node which is immediately upstream to the disrupted link (named pivot node or helped node) starts to search for an alternative path by broadcasting a Help Request (HREQ) message [55]. Every neighbor node receiving this message does the following comparisons sequentially depending upon the results of the previous step.

a) The message will be discarded if it is found that the downstream node of the receiver node is the escaped or the helped node because the motto of seeking any other path is to avoid the escaped node.

b) The message will be discarded if it is a duplicate message i.e. the node has already received the same HREQ packet.

c) If the comparison results do not fall in (a) and (b), then the cost of the escaped node is compared with that of the current node by the node itself. If the current node’s cost is found equal to or lesser than the escaped node, it is believed that this route won’t lead to an optimal path and a Help Response (HREP) message to the helped node is sent. Otherwise, the node routes the HREQ message to the next downstream node, and above mentioned three steps in a to c are repeated.

d) The candidate node with minimal cost will be selected as the alternative path by the helped node, and the routing table shall be updated accordingly and, the data packet is routed to its new downstream node.

3) ROBUST DISJOINT MULTIPATH (RDM) ROUTING ALGORITHM

Authors have developed a Robust Disjoint Multipath (RDM) [56] routing hole detouring approach to identify holes and keep the network in working condition despite holes. In multipath routing, there can be data collision on hole boundaries as data is concentrated on borders of the hole and the network may undergo too much energy consumption. Hole size may increase due to this energy-burning up and congestion in the network. Bigger the hole, the more problematic it is, also known as the hole growing effect. Disjoint multipath routing has been used for designing a hole detouring scheme to forward data from the source node to the destination node and solve issues. It eliminates flooding overhead to determine multi-paths.

It is known that the sensor nodes in the region $\alpha^2$ follow the mathematical distribution function [57] given that the wireless sensor network is having a distribution function with density $\rho$. Authors have supposed that all nodes in the network are aware of their location, the location of the destination node and transmission route is comprised of multiple hops and the network has a 2-D (two dimensional) Gaussian function. The network has been divided into several virtual grids to disseminate data. Every virtual grid contains parameters: reference coordinates $(x_r, y_r)$, grid size $\alpha$. One or more sensor nodes lie in area A having a minimum value of $\alpha$, where A is $\alpha^2$.

Different paths to the destination are determined by the source node. A continuum model has been introduced by authors to disjoint these multiple paths. There is a pipeline from the source sensor node to the destination node in which
the intermediate point is called the anchor point. Even the smallest pipeline $P_0$ consists of a continuum of the $\alpha^2$ area comprising one or more sensor nodes. Disjoint paths are designed from source to destination and are rebuilt to detour holes for reaching the destination. When any hole is encountered, data transmission is toggled into perimeter mode. There exist multiple paths with pipelines $P_k$ for $k = 1, 2, \ldots, N$. If a hole is found in one pipeline then another pipeline is used bypassing the hole by avoiding virtual grid with hole. Rules to build disjoint multiple paths for hole detouring scheme are as follows:

1) For any pipeline $P_k$ having an odd value of $k$, the right-hand rule is followed on the hole boundary for hole detouring.

2) For any pipeline $P_k$ having an even value of $k$, the left-hand rule is followed on the hole boundary for hole detouring.

3) If multiple channels are there at the boundary of a hole, a packet for detecting hole boundary is sent in order as follow:

\[
P_k > P_{k+1} \quad \text{for any } k \geq 0
\]

4) In destination mode, if any hole is met during packet forwarding from source to destination, the approach is changed to perimeter mode.

5) When multiple paths send data packets to meet the same hole concomitantly, the hole is passed by hole boundary packets as in step 3) and the paths wait to send data packets respectively.

4) PRIORITY BASED MULTIPATH ROUTING ALGORITHM (PBMRRA)

Authors have proposed a Priority Based Multipath Routing Algorithm (PBMRRA) [58] to identify holes and endow them with high energetic multi-paths. The algorithm comprises three major components (shown in Fig. 11): one section identifying neighbor nodes; another section to construct multi-paths using selected neighbors; and last section to make a priority-based selection of high energy routes using metric path delay, liveliness (residual energy), and hop count (number of hops amid source and sink node).

Sensor network has been presented in form of a graph where sensors represent nodes in the network and edge is determined using delay d, energy level e of node, and hop count hc. These values d, e, and hc are updated periodically for every node.

After initially deploying the network, every sensor node calculates its neighbor nodes using the Neighbourhood Selection Algorithm (NBA). Using NBA, every node sends a message to neighbor nodes to build its neighborhood. Nodes with required transmission power and energy level [59] greater than a threshold are selected as neighbor nodes. Quality of link to neighboring nodes [60]–[62] is calculated by the sensor node. The quality of the link [63], [64] varies with time.

After establishing a neighborhood using NBA, multiple paths are determined, and the best paths are chosen out of elected paths for data transmission using Multipath Construction Algorithm (MCA). The sink node initiates the path discovery process for every source node. A request control packet is sent by the sink to the source node through intermediate nodes. Source node replies with a reply message and after that multipath construction process is started. Three values e, d, and hc are updated by every node and are cross verified with neighbor nodes. Sensor nodes having energy levels greater than and delay less than the threshold value are selected by varying values of threshold.

After electing multiple paths [65], the Priority Selection Algorithm (PSA) is applied to prioritize paths based on values of d, e, and hc. PSA is a predictive algorithm that considers thresholds for prioritizing paths. The priority value for each chosen path is computed using values of d, e, and hc. Path with uppermost priority is a route with maximum energy and the next priority path also has the next energy level and henceforth from the source node to sink node [66]. Paths with very low energy are repudiated [67].

Once routing paths are out of order and stop working. PBMRRA identifies holes in the network using the energy level and status of nodes and broken paths. The algorithm provides small delay and better throughput for the small-sized network but delays upsurges with an increase in the number of nodes.

5) GREEDY PERIMETER STATELESS GEOGRAPHIC ROUTING (GPSR) ALGORITHM

GPSR [68] is a geographic routing technique that makes routing decisions using greedy forwarding by using only the
FIGURE 12. An example showing weak stuck node u. Here, a solid circle represents node u’s transmission range.

immediate neighbor’s position/coordinates in the network. Every node knows the geographic locations of neighboring nodes and makes the optimal decision to choose the next hop looking that which neighbor is the geographically closest to the destination node. Such forwarding process seeks in succession closer geographic nodes till the destination is not reached. To know the location and be in synchronization with its neighbors, a simple beaconing algorithm is used, wherein, each node broadcasts a beacon containing its own ID and position. If a beacon is not received from any neighbor node in timeout interval, the node updates its neighbor list and considers that neighbor has either failed or gone out-of-coverage range. Sometimes greedy forwarding fails because of routing holes created due to local minimum (Figure 9). Recovery to failure is done by using perimeter routing mode i.e. when a data packet reaches an area where further movement is unfeasible, it is routed around the perimeter of that region. The right-hand rule is applied in perimeter routing which says that when reaching from a node x at node y, the edge which is sequentially counterclockwise about y from edge xy will be traversed next. The sequence of edges traveled by the right-hand rule is called a perimeter. For the right-hand rule, it is required that no edges cross each other. Taking this into consideration, GPSR uses either Gabriel Graph (GG) or Relative Neighbourhood Graph (RNG) [69], [70] to obtain a planar graph of a network having all non-crossing edges. Removing edges from the graph that are not part of the RNG or GG would result in a network with no crossing links.

6) TENT RULE AND BOUNDHOLE DISTRIBUTED ALGORITHM

Many routing algorithms use greedy forwarding techniques to transmit data packets to their destinations in sensor networks [71]–[75]. In this approach, authors have tried to overcome ‘the local minimum phenomena’ which can be one of the causes of packets getting stuck. Nodes, where packets may get stuck in a greedy multi-hop forwarding technique, are defined as stuck nodes. Stuck nodes are further categorized into two types: weak stuck nodes and strong stuck nodes.

A node u∈ S is defined as a weak stuck node if there is any such node v∈ S outside u’s transmission range and there is no 1-hop neighbor node of u that is closer to v than u itself. Any such destination node like v is referred to as a black node of the sender (here, it is node u), as shown in Fig. 12.

To find weak stuck nodes, the Voronoi diagram and the Delaunay triangle are used. In the Voronoi diagram, the whole plane is divided into small convex regions in such a way that all the points lying in a region are closest to only one node; such convex regions are referred to as Voronoi cells. The Delaunay triangulation represents the dual graph of the Voronoi diagram; it connects those nodes which are present in adjacent cells in the Voronoi diagram. In the Delaunay triangulation DT(S), a node p is said to be a stuck node if there is no edge adjacent to node p which is far from more than a 1-hop neighbor node. Restricted Delaunay Graph (RDG) is generated to find holes generated using weak stuck nodes; it is subgraphs of the Delaunay triangulation in which only those edges which are not longer than 1 are kept. A hole is said to be a face in the RDG having at least 4 vertices and all the weak nodes on the boundaries of the hole.

A node u∈ S is said to be a strong stuck node if there is a point that is outside u’s transmission range and there is no 1-hop neighbor of u which is closer to the point than u itself. The gathering of such locations is known as the black region (shaded area shown in Fig. 13).

The TENT rule (Fig. 14) has been used for detecting the stuck nodes. The steps followed are:

a) For each node u, first, all the 1-hop neighbors of node u are ordered counterclockwise.

b) A perpendicular bisector is drawn for every pair of nodes that are adjacent neighbors (shown in Fig.14). Bisectors drawn using ux and uv dissects at point O.

c) If point O lies in the communication range of node u, then there is a black region lying in the communication range of u.
C. SECURITY HOLES IDENTIFICATION TECHNIQUES

Addressing the protection and defense mechanisms of WSNs, it is important to address an issue related to security holes. These holes have been divided into various categories depending on which layer the attacker chose to affect the network. Major security holes are Jamming holes, Blackhole, Sinkholes, and Wormholes.

Jamming holes [15], [76], [77] are unlike other categories of holes encountered in the sensor network such that you twit communication/sensing capabilities of nodes. Various spread spectrum techniques are generally used for radio communications to handle jamming issues, but cost and complexity factor plays an important role in adopting any of the techniques. No such technique is available, yet which has effectively counter jamming with remarkable results. There can be cases in tracking applications where the target is furnished with jammers having the capability of jamming the radio frequency which is being used for communication amid the sensor nodes. In this case, the presence of the target will be sensed/detected by other nodes in the surrounding, but they won’t be able to communicate it back to the sink due to the communication jamming in the region. This region influenced by jamming centered around the jammer is called a jamming hole. This may be the case of purposeful or unintentional jamming. In purposeful jamming, a bad node/an adversary intentionally creates a denial-of-service state. Unintentional jamming occurs due to malfunctioning of nodes that are not behaving properly and start continuous transmission and hence occupies the whole wireless channel refuting the services to other neighbor nodes. Besides communication jamming, there are also possibilities of jamming sensing capabilities in a few special cases e.g., if acoustic sampling is being used by the sensor network for tracking objects and objects which are getting tracked starts introducing random and high-power acoustic noises, it would not be feasible for the sensor nodes to discover the presence and report the existence of the object.

An adversary node in the sensor network can easily introduce Sinkhole (shown in Fig. 15(a)) attacks is a way opted by an attacker to illustrate compromised sensor node as a very attractive route by publicizing very cost-effective and catchy routes [15], [78]–[80]. As the malicious node appears attractive, it may be selected as the next hope by neighbor nodes considering its best route, moreover, this route gets circulated to other nodes as well. In this way, most of the traffic is attracted toward malicious nodes, and the node can drop the packets, selectively transmit or change the content of the message. Thus, the malicious node creates a sinkhole centered around itself. It will lead to more energy consumption and result in congestion among the nodes involved, which will lead to node failure due to power shut down and, hence, the creation of routing holes. The possibilities of denial-of-service attacks increase due to sinkholes presence in a sensor network, and the number of other types of denial of service (DoS) attacks also increases. A black hole (shown in Fig. 15(b)) is a type of attack [81] in which a node drops all the packets it receives.

Wormhole [59] is a type of attack in which malicious nodes located poles apart in the WSN, form a “tunnel”. Malicious nodes receive data from one point in the network and replay them in another part of the network through the tunnel via a different radio channel. Other nodes located in the network poles apart assume that they are neighbors which lead to a wrong routing convergence.

Major techniques used to identify Jamming, Sink, Black, and Wormholes (shown in Fig. 16) are as follows: -

---

**FIGURE 14.** Graphical representation of the TENT rule. Here, the solid circle is representing node u’s transmission range [47].

---

d) Going in counterclockwise order, nodes x and y are adjacent to each other. As no nodes are lying in the black region, hence node u cannot get stuck for any destination which is inside the cone created using ux and uy.

e) On the contrary, if the center O is falling outside the communication range of u, any such destination must be there in the plane where the packet will get stuck at u.

f) A node having one or more stuck angles is said to be a stuck node. It is defined that the node u cannot get stuck if the angle xuv is smaller than 120 degrees.
TABLE 3. Routing hole detection – advantages and constraints.

| Researcher       | Model/ Concept Used                          | Constraints                                                                 | Advantages                                                                                   |
|------------------|----------------------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Intanagonwiwat et al.[54] | Directed Diffusion (DD) Single Path Routing Algorithm | All Nodes need to be data-aware. Flooding of Data is more. | It uses two different mechanisms to deal with different application requirements. The protocol can also behave as a multipath protocol by keeping alternate routes active through a timely reinforcement. |
| Tian et al.[55] | Pivot-initiated path repairing Single Path Routing Algorithm | In this approach, each intermediate forwarding node needs to have the means to get feedback about the delivery status of data sent to the successor node. | The approach has a good delivery ratio and seems more energy efficient. |
| Kim et al.[56] | Robust Disjoint Multipath (RDM) Routing Algorithm | Sensor nodes need to be aware of their locations. Nodes sending data packets must know the location of the destination node. | The protocol reduces the overhead of reconstructing routes and flooding. Energy consumption and signaling overhead are less. |
| Suseela et al.[58] | Priority Based Multipath Routing Algorithm | Choosing the right threshold values for the metric delay, energy level, and hop count is difficult. Network scalability is poor. Delay increases with an increase in network size. | The algorithm provides reliable paths and good throughput. |
| Karp et al.[68] | Greedy Perimeter Stateless Geographic Routing Algorithm | All nodes need to know the geographic location of their neighbors. To upkeep planar graph at every node adds extra complexities as well as cost. | The protocol has good scalability towards increasing numbers of nodes in the network. |
| Fang et al.[71] | TENT rule & BOUNDHOLE Distributed Algorithm | The complexity of the algorithm increases with the increase in the number of nodes. | It overcomes the problem of local minimum phenomena which generally occurs in greedy forwarding. |

1) JAM PROTOCOL
Jamming is a way to interfere with the capability of an adversary to communicate [78]. Few well-known types of jamming can be recognized by a receiver through their unique energy patterns. JAM [15], [76] detects and maps jammed areas in a sensor network. Jamming detection and mapping modules are two prime components of the JAM mapping service. To make a distinction between jamming and normal interference, the JAM detection module monitors radio traffic and MAC layers and subsequently applies different heuristics, e.g., bit-error rates, etc., depending on the data available and the result of that is compared with a threshold value, if it is above threshold then the interference is declared as jamming. But, while deciding the threshold value, determining threshold value is a very critical task. Whenever jamming is detected by any node, a broadcast message called JAMMED is sent to its neighbor nodes with high priority overriding carrier sense mechanism generally enforced by MAC layer. Jammed area mapping is started by nodes which are neighbors of nodes present in the jammed area and have got a JAMMED message by a node lying in the area covered by a jamming hole.

Receiving nodes wait for multiple JAMMED messages for a short time period before sending a Build message to reduce multiple broadcasts with partial information. When the short timer expires, a BUILD message is transmitted by the receiving node to its neighbors, and the process of creating the boundary of the jamming hole is started. When any node receives a BUILD message from an unknown group, the corresponding node creates that group locally and stores the information otherwise, if the message is from a known group; it is updated in the list. Each unique build message is identified by its originator and sequence number, this helps in preventing duplicity of the messages. Boundaries of holes are formed by combining multiple BUILD messages originated from the neighbor nodes.

2) RSSI AND PDR BASED DOS DEFENSE MECHANISM
For identifying jamming attacks, authors have mentioned three techniques which comprise statistical analysis of the received signal strength indicator (RSSI) values, the packet delivery ratio (PDR), and the carrier sense time (an average time required for sensing an idle channel) [82]. All three techniques need a few baseline measurements. The use of only these techniques is not enough to recognize jamming; some algorithm is also required in addition. There are many such available algorithms. In one algorithm, first of all, poor link utility is identified by PDR analysis first and then RSSI analysis is used as a consistency check to find out whether the poor network performance is being created as a result of jamming. In another approach, to defend against jamming, the jammed region is collaboratively identified by the nodes and then traffic is routed around it. If a jamming attack is identified by any sensor node, then one way out to defend itself is to set a long-term sleep mode for sensors nodes and wake them up at regular intervals to check if the channel is still...
Though this may not stop a DoS attack [83], it could considerably increase the life span of sensor nodes by curtailing energy consumption. Thus, there will be a reduction in the number of nodes failing due to jamming as the attacker would have to jam for a significantly longer time span, most likely leading to power exhaustion before the targeted nodes do, so. On a similar concept of analyzing incoming frames and delays, algorithm for selective jamming issue has also been developed by researcher for IEEE 802.15.4 beacon-enabled mode WBANs (Wireless Body Area Networks).

3) GRASSHOPPER OPTIMIZATION INTRUSION DETECTION SYSTEM
Authors have proposed a grasshopper optimization algorithm (GOA) [84] based intrusion detection system named GOIDS (Grasshopper Optimization Intrusion Detection System) to alleviate and diminish DoS (Denial of Service) attack. Machine learning-based GOA has been implemented along with support decision tree (C4.5), vector machine (SVM), naïve Bayes (NB), and multilayer perceptron (MLP) to optimize the performance of the technique.

Grasshopper Optimization Algorithm is a meta-heuristic algorithm that works through inspiration from the life span of grasshoppers. The life of a grasshopper has been mainly divided into two phases; in the beginning, swarms grow slow, i.e., shorter steps and in the second phase, development is fast, i.e. longer steps [85], [86]. Small steps in the first phase lead to exploration and the second phase takes process toward exploitation.

Binary grasshopper optimization has been given. To deal with the binary feature selection problem, the algorithm uses a transfer function to convert continuous feature selection into binary [87]. After choosing the best possible feature subset, data is passed for training to the classifier. Authors have used different classifiers and techniques C4.5 has been observed as a better fit among other classifiers. Distributed DoS attacks are attacks where multiple infected nodes act as attackers. The proposed model (shown in Fig. 17) is run to observe incoming traffic and identify attacks. Along with good accuracy and performance, model both load and traffic in the network.

4) OAO DV INTRUSION DETECTION SYSTEM
It Observed AODV (Ad hoc On-Demand Distance Vector) [88] had been designed as a countermeasure against one of the most devastating attacks that are the black hole attack. (Observed AODV) has been proposed as a secure extension of the AODV protocol. AODV is a very energy-efficient protocol and requires very little computing power and is hence considered very suitable for sensor nodes having limited capabilities. An attacker can show itself as promising during the route discovery phase by falsifying RREQ and RREP control packets or during the data packet transmission phase by storing data in transit and deleting afterward. The network is portioned into clusters to apply OAODV and cluster heads for every cluster are identified. Protocol mainly uses two components-

- Intelligent Watchdog – ensure that packets are properly forwarded, and retransmitted packets also meet AODV specifications.
- IABO (Intelligent Activity-Based Overhearing) – continuously monitor traffic from neighbour nodes.

The cluster heads choose observer nodes in the cluster and coordinate the activities of observer agents. These observer agents in the protocol are in charge to monitor data flow in their neighborhoods and analyzing inconsistency in order to detect mistrustful activities of neighbor nodes if any. A concept of penalty and reward has been introduced in the algorithm to evaluate the trust level of nodes. This evaluation of trust level leads to determining, isolate and detecting
5) JENSEN–SHANNON DIVERGENCE BASED INDEPENDENT COMPONENT ANALYSIS (JDICA) TECHNIQUE

Jensen–Shannon Divergence Based Independent Component Analysis (JDICA) technique [89] has been proposed by authors to identify black holes with good detection accuracy. JDICA works by analyzing the behavior of sensor nodes through trust factor, energy, and cooperation and finding out interdependence amid sensor nodes using mutual probability function and probability distribution function based on Jensen–Shannon Divergence.

A linear transformation has been achieved through independent component analysis. Through this, multivariate signals have been separated into smaller additive components. Jensen–Shannon divergence determines mutual information among sensor nodes and hence independence. Various factors are determined as below:

- Energy Usage – It denotes energy consumption and is calculated by determining the total quantity of energy used up in receiving and transmitting data.
- Trust – It is measured from data drop rate, i.e., from the difference between data packets forwarded and dropped out of total received data.
- Cooperativeness – It is a measure of how actively a node participates in transmission i.e., from link estimation.

Using the above factors, the JDICA model (shown in Fig. 18) identify holes by calculating variance amid sensor node and black hole. Malicious nodes are isolated from the network through the broadcast of messages to nodes in the network and hence results in raise in packet delivery ratio.

6) ARTIFICIAL BEE COLONY-ATTACK DETECTION (ABC-AD) ALGORITHM

Authors have proposed an artificial bee colony algorithm-based hole detection mechanism [90] for detecting sinkholes in the network. The algorithm creates a table of rule sets and works based on rule matching methodology. The model comprises of four modules:

- Local Packet Monitoring
  The purpose of this module is the collection of data and audits by listening to messages of neighbor nodes.

- Local Detection Engine
  This module consists of a rule set defined to identify sinkholes. A rule set is prepared in tabular format in which each sensor node’s information about neighbor nodes and respective link quality is stored. When the request for an update in route is received, the node compares link quality of received request with already
stored in the ruleset, if they match, then the route is updated. If there is a mismatch, then it may result in a sinkhole. Local detection engine passes such suspects which have the probability to be sinkhole to cooperative detection engine.

- **Cooperative Detection Engine**
  The module evaluates suspects and generates alert regions.

- **Local Response Module**
  The module identifies intruder sinkholes.

The algorithm works on rule matching and whenever any route update request is received, it is checked for a match with already stored value. When there is a mismatch, the node is declared a sinkhole.

7) **HOMOMORPHIC ENCRYPTION & WATERMARKING BASED TECHNIQUE**

Authors have proposed homomorphic encryption and watermarking-based technique [91] to identify sinkholes and can identify the tempering of data, therefore, making certain data integrity. The model is lightweight and consumes low energy, hence leading to an increase in the lifetime of the network. It follows a cluster-based communication mechanism where nodes in one cluster transfer data to their cluster head and cluster head communicate with sink/destination node. For routing, the model uses TEEN (Threshold Sensitive Energy Efficient Sensor Network) protocol [92]. The powerful sensor node is a cluster is chosen as cluster head as per TEEN protocol criteria selection mechanism. Once cluster head is chosen, it broadcasts two types of threshold values as below every time at cluster setup phase:

- **Hard threshold**: If this value of sensed feature is realized by any node, it needs to unlock the transmitter and report for the same to the cluster head.
- **Soft threshold**: It is little deviation in the sensed feature’s value, due to which node is unlocked.

Nodes constantly watch their neighbor nodes, whenever any situation is there that node finds out sensed value has reached soft/hard threshold, it reports to cluster head. The nodes observe their surroundings continuously. The TEEN protocol makes use of data aggregation to lessen data transmission time [93] and boost data robustness and accuracy.

Watermarking and homomorphic encryption technique (shown in Fig. 19) has been proposed to prevent sinkholes and
ascertain the authenticity and integrity of data. The technique works in four phases. These are:

**a: Initialization phase**
In the initialization phase, different keys; keys for cluster heads, and network keys are generated by the base station. The cluster key is used for intracluster communication whereas the network key is used for inter-cluster communication. Paillier cryptosystem [94] has been used for key generation. To avert alteration in keys by the intruder, a lightweight homomorphic private key encryption technique is used by the base station.

**b: Sensing phase**
Watermarking is used by sensor nodes that provide copyright/ownership, ascertain confidentiality and integrity of data. Watermarking technique uses Pseudo Generated Random Number (PGRN) [95] and Hash-Based Message Authentication Code (HMAC) [96] for watermarking. The technique doesn’t require any storage space which makes it more suitable for small-sized sensor devices.

**c: Cluster head verification and data aggregation phase**
Whenever any packet is received by the cluster head, it extracts the packet’s timestamp and verifies if it is a new or old packet. Old data packets are discarded and information about the node sending data is reported to the base station node. The authenticity of new packets is verified by extracting watermarked data. Cluster head aggregates data from multiple nodes, encrypt IDs of these sensor nodes, apply watermark and send the data to the base station node.

**d: Base station verification and confirmation phase**
When the base station receives data from cluster heads, it extracts timestamp, encrypted IDs, and watermarked data from received data. The base station also drops data with an old stamp. Extracted watermarked data with a new timestamp is compared with regenerating watermarked data using an assigned key to that cluster head node. If data doesn’t match, the ID of this attacker cluster head is propagated in the network and it is blocked by the base station.

The proposed model is robust against deceiving attacker nodes as it uses encrypted Node IDs, averts any tempering in data through watermarking, and prevent replay attack by dropping old packets.

8) SUSPICIOUS LINK DETECTION AND WORMHOLE VERIFICATION IN OLSR
Authors have introduced a technique to discover and curb wormhole attacks in OLSR [97]. First, suspicious links which may form part of wormhole tunnels are detected, and then an appropriate wormhole detection method is used to distinguish wormhole links from other legitimate links. In the detection mechanism, encrypted probing packets are exchanged among the two speculated neighbors (ends of the wormhole) to make a distinction.

**Detecting Suspicious Links:** In OLSR, a HELLO message is broadcasted by every node on a frequent basis to find out its one-hop neighbors. Nodes receiving HELLO messages revert a message to the sender as a neighbor. This HELLO message can be replayed from the sender node to the nodes more than one hop away in a wormhole attack. In this approach, authors first discover network links that are more prone to wormhole attacks, and then a wormhole detection scheme is applied. One of the basic detection strategies to detect wormholes is to observe the latency as wormholes have longer packet latency in comparison to the normal wireless propagation latency on any hop. In any wormhole attack, propagation latency is considered a major contributing factor. As basic nature of wormhole, many multi-hop routes are attracted toward wormhole, thereby, overloading any particular multi-hop route. Overall as a result of overloading, longer queuing delays get incorporated in the rate. Yet this is a way of detection but still not a sufficient condition to find out wormhole, because latency in packet transmission may be due to various other reasons also like congestion and intra-node processing delays. Hence latency/delays alone cannot be considered as wormholes but the links which possess longer delays are marked as suspicious. Now, Wormhole verification is done on the links which are marked suspicious. For the OLSR protocol, two new control packets are defined: HELLO_{req} and HELLO_{rep} to analyze these links. The HELLO message in...
OLSR is superseded by the HELLO$_{req}$ message; the message can be depicted as any of the two discussed above depending on the selection of choice. It performs the function of the original message in the standard option whereas in another case, the HELLO message is used to get an explicit reply with a HELLO$_{rep}$ message from its neighbors. When any node gets a HELLO$_{req}$, it notes down the sender node’s address, and the time $D_i$ left until it is scheduled to send its next HELLO message. To avoid overloading the network with too many HELLO replies, replies of multiple requests are piggybacked to the normal HELLO message by the receiving node. For each piggybacked reply, the sender’s address and values of $D_i$ corresponding to HELLO$_{req}$, are recorded by the node appended to the message. When a HELLO$_{rep}$ message is received by any node, it is checked whether any information regarding any of its pending requests is contained in this HELLO$_{rep}$. If no such information is found, then the received HELLO$_{rep}$ is treated as a standard HELLO message. If the message is not found as a normal/standard HELLO message, then the arrival time of HELLO$_{rep}$ is checked to know if the time taken by the message is within its specified timeout period after taking into consideration the delay $D_i$ occurred at the receiver. If it is found that HELLO$_{rep}$ has reached within its scheduled timeout interval, then the initiator node marks the communication link connecting itself to the node which has sent HELLO$_{rep}$ message as safe, otherwise the link is marked as suspicious and any sort of communication with that node is stopped until the wormhole verification process is completed.

After detecting suspicious links, verification is done by the originator node to find out whether the path connecting itself to other ends of the suspicious links is not any wormhole tunnel. Two new message types have been used for this purpose: a Probing packet and $\text{ACK}_\text{prob}$. A Probing packet is sent to all of the suspect nodes. Nodes that receive the Probing packet, reply with an $\text{ACK}_\text{prob}$ message telling its own wormhole status as well as processing time details to the node which is the originator of the Probing packet.

The reputation state of a node can be either “proved” or “suspicious” based on the results inferred using the previous exchange details of HELLO$_{req}$ and HELLO$_{rep}$ messages during the suspicious link detection process. This timing information given by $\text{ACK}_\text{prob}$ is used to figure out the timeout period accurately. If the node receiving the Probing packet is not aware of the state of the originator of the packet, it ceases to send the $\text{ACK}_\text{prob}$ and starts gathering the required details through the exchange of HELLO$_{req}$ and HELLO$_{rep}$ messages. To secure the communication, end-to-end authentication is kept during the exchange of the Probing packet and $\text{ACK}_\text{prob}$. Similar to HELLO$_{Req}$ and HELLO$_{Rep}$ exchange process, it is verified by the originator node of the Probing packet that the $\text{ACK}_\text{prob}$ has been received within the specified timeout period or not. Based on the verification, possible suspicious links are checked. A comparative analysis is done to see an evaluation of the reputation of the other endpoint in the suspicious link to the other node’s evaluation to check whether any suspicious link is creating a wormhole tunnel:

(Proved, Proved): If the status of the remote node as well as the contents of the encrypted $\text{ACK}_\text{prob}$ is proved, it is concluded by the originator node that the link amid itself and the suspicious node is safe and hence won’t be creating a wormhole tunnel. Therefore, a neighborhood relationship will be maintained by the originator with the node by sharing as well as accepting data from that node.

(Suspicious, Proved) or (Proved, Suspicious): If it is observed by one of the two nodes that the remote node or content of $\text{ACK}_\text{prob}$ is suspicious, it is declared by the originator that the link is still suspicious. Communication with that node is stopped and will be restarted after a random interval of time followed by an exchange of Probing and $\text{ACK}_\text{prob}$ packets. After this exchange of packets, it is found that there is at least one suspicious state then this link will be treated as a wormhole tunnel by the originator from that particular point onwards.

(Suspicious, Suspicious): If both nodes set the status of the remote node and the contents of the $\text{ACK}_\text{prob}$ as suspicious, the link is marked as a wormhole tunnel by the originator, and this node is removed from the list and further use of this link is stopped completely by dropping the packets which are using this link as a medium if any until the next HELLO$_{req}$-HELLO$_{rep}$ exchange process takes place.

9) ENERGY PRESERVING WORMHOLE IDENTIFICATION IN AODV

Authors have designed a wormhole detection technique in the ad-hoc on-demand distance vector routing (AODV) [98] which works using connectivity information of the network. The algorithm is used during route discovery when the route reply packet is received by the sender [99]–[104]. The algorithm is started from the destination point and the detection process is run with neighbor nodes in the elected path (shown in Fig. 20) in two stages.

- The first stage is to construct a list of one-hop neighbors of sensor node A and that of sensor node B. After constructing these lists, it is checked if there is any common node amid two. If yes, then nodes are not intruders, if there is no common node, the algorithm runs for the second stage.
- In the second stage, it is checked whether there is any common node between a two-hop neighbour list of node B and a one-hop neighbour list of A. If yes,
| Researcher                | Hole Category | Model/ Concept Used                                      | Constraints                                                                 | Advantages                                                                 |
|--------------------------|---------------|---------------------------------------------------------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Wood et al. [76]          | Jamming Hole  | JAM: A Jammed-Area Mapping Service for Sensor Networks  | It is assumed that each node knows its location and has also been assigned a unique ID. Choosing a threshold beyond which the interference needs to be treated as jamming is a very critical exercise. A carrier-sense defeating broadcast method is required to report the jamming. The JAMMED message may not reach any neighbor outside the jamming region in case of continuous channel jamming. | The protocol is robust to failure rates.                                     |
| Raymond et al. [82]       | Jamming Hole  | RSSI and PDR based DDoS Defense Mechanism                | A threshold needs to be set for signal strength, packet delivery ratio.      | Identify jamming and save the network from going down.                      |
| Dwivedi et al. [84]       | Jamming Hole  | Grasshopper Optimization Algorithm                      | Choosing exploration and exploitation phase step size plays a major role in deciding whether the algorithm will lead to global optima or not. The algorithm doesn’t give deterministic results. | The model gives results with good accuracy and reduces the load on the network. C4.5 classifier-based learning has proved the best fit among all classifiers. |
| Elie T. FUTE et al. [88]  | Black Hole    | ODAV Intention Detection System                         | The technique is applicable only for homogeneous and static sensor networks. Also, partitioning of the network into clusters is required where each cluster is headed by a sensor node called cluster head. | As AODV is a simple protocol and requires less computing power, and OADV is an extension of AODV so it is an energy saver, doesn’t introduce much delay, and has good network throughput. |
| Sundar et al. [89]        | Black Hole    | Jensen–Shannon Divergence Based Independent Component Analysis (DICA) Technique | The algorithm considers that sources taken for component analysis are linearly independent. | The algorithm isolates the black hole after detection which leads to an increase in successful transmission of data, with reduced delay. Algorithm detection time is comparatively low. The false-positive rate is very low. |
| Nithyanandam et al. [90] | Sink Hole     | Artificial Bee Colony - Attack Detection (ABC-AD) Algorithm | The algorithm will take time to become stable as bees collectively work towards the result, and thus complexity will increase with an increase in network size. Any tempering of data will lead to false results. Also, it doesn’t give deterministic results. | Energy consumption is low and packet delivery is improved. The algorithm possesses less time complexity. |
| Ibraheem et al. [91]      | Sink Hole     | Homomorphic Encryption and Watermarking based Technique  | Clustered WSN requires a few powerful nodes which can compute all the keys and distribute them in the network. Sensor nodes need computational power to run a multi-part algorithm. | The technique is energy efficient, fast, and provides good throughput. The protocol is lightweight and certifies data integrity during transmission. |
| Nait-Abdesselam et al. [97]| Worm Hole     | Suspicious Link Detection and Worm Hole Verification in OLSR | The request-reply process is very complex, shall consume very much power, and doesn’t seem suitable for larger networks. | Location information of node or any time synchronization is not needed in this method, and its detection rate is good. |
| Aliady et al. [98]        | Worm Hole     | Energy preserving Worm Hole identification in AODV       | The algorithm needs network connectivity information. The algorithm may not always be able to identify short-length wormholes. | The technique is energy efficient. It doesn’t add any extra cost as no specialized hardware or computational capability is required. A wormhole tunnel of four or more hop lengths is detected with a quite good accuracy. |
TABLE 5. Summary for coverage hole detection techniques/approaches.

| Researcher          | Model/Concept Used               | Node Deployment Strategy | Sensing Model                      | Coverage | Computation Strategy | Major Parameter | Tool                                      | QoS Parameters                                      |
|---------------------|----------------------------------|--------------------------|------------------------------------|----------|---------------------|-----------------|------------------------------------------|-----------------------------------------------------|
| Wang et al. [22]    | Voronoi Diagram                  | Random                   | Isotropic, heterogeneous           | 1-coverage | Distributed         | Location        | NS-2 (version 2.1b9a)                    | Coverage, Convergence Time & Energy Consumption     |
| Zhao et al. [28]    | DSCS and DW Algorithm            | Random                   | Isotropic, homogeneous             | 1-coverage | Distributed         | Location        | MATLAB 2013b                             | Correct and False Detection Rate, Energy Dissipation, Runtime, and Algorithm Complexity |
| Huang et al. [29]   | k-UC and k-NC polynomial-time algorithm | Random                   | Isotropic, homogeneous & heterogeneous (both) | k-coverage | Centralized         | Location        | Author developed Simulator Toolkit       | Coverage Level, Algorithm Complexity                |
| Silva et al. [32]   | Blind Swarm                      | Random                   | Isotropic, homogeneous             | 1-coverage | Centralized         | Range-based - Strong or Weak Signal | MATLAB, Plex & CHomP | Coverage                               |
| Beiramao et al. [37]| Two Anchor (TA) and Cyclic Segment Sequence (CSS) | Random                   | Isotropic, heterogeneous           | k-coverage | Distributed         | Range           | ---                                      | Error Index, False Positive, Coverage               |
| Corke et al. [17]   | Unmanned Aerial Vehicle (UAV)    | Uniform                  | Isotropic, homogeneous             | 1-coverage | Centralized         | Connectivity    | Hardware and software from these labs: USC's helicopter, Dartmouth's sensor network, and CSIRO's interface among sensor network and helicopter at USA | Field experiments executed at USC Lab, USA for Connectivity and Deployment Error measurement |
| Yan et al. [44]     | Czech complex and Rips complex   | Random with Poisson Distribution | Isotropic, homogeneous             | 1-coverage | Distributed         | Connectivity    | MATLAB and Jplex Package                | Coverage Hole, Error Probability                   |
### TABLE 6. Summary for routing hole detection techniques/approaches.

| Researcher          | Model/ Concept Used                          | Node Deployment Strategy | Sensing Model          | Routing      | Computation Strategy | Major Parameter                          | Tool     | QoS Parameters                                      |
|---------------------|---------------------------------------------|--------------------------|------------------------|--------------|----------------------|------------------------------------------|----------|-----------------------------------------------------|
| Interangono/wat et al. [54] | Directed Diffusion (DD) Single Path Routing Algorithm | Random                   | Isotropic, homogeneous | Single Path  | Distributed           | Interest & Gradient                      | NS-2     | Average Disipated Energy, Average Delay, Event Delivery Ratio |
| Tian et al. [55]     | Pivot-Initiated Path Repairing Single Path Routing Algorithm | Random                   | Isotropic, heterogeneous | Single Path  | Distributed           | Help Request and Response Messages       | ---      | Energy Consumption, Node Failure Ratio, Channel Error Rate, Delivery Ratio |
| Kim et al. [56]      | Robust Disjoint Multipath (RDM) Routing Algorithm | Random with Gaussian Distribution | Isotropic, homogeneous | Multi-Path   | Distributed           | Location                                | Qualnet 4.0 | Signalling Overhead, Energy Consumption, Average Delivery Delay, Accuracy |
| Susaeta et al. [58]  | Priority Based Multipath Routing Algorithm   | Uniform                  | Isotropic, homogeneous | Multi-Path   | Distributed           | Location, Threshold, Hop count           | NS-3     | Throughput, Received Rate, Time Complexity          |
| Karp et al. [68]     | Greedy Perimeter Stateless Geographic Routing Algorithm | Uniform                  | Isotropic, homogeneous | Geographic   | Distributed           | Location                                | NS-2     | Packer Delivery Success Rate, Protocol Overhead      |
| Fang et al. [71]     | Tent Route & Boundhole Distributed Algorithm  | Random but Uniform       | Isotropic, homogeneous/ heterogeneous | Geographic   | Centralized           | Location, The authors developed Simulator using C++ |         | Region of Interest, Path Migration                   |
TABLE 7. Summary for security hole detection techniques/approaches.

| Researcher                  | Hole Category | Model/Concept Used                        | Solution Category | Node Deployment Strategy | Sensing Model | Layer/Protocol | Computatio n Strategy | Major Parameter          | Tool                          | QoS Parameters                  |
|-----------------------------|---------------|-------------------------------------------|-------------------|--------------------------|---------------|----------------|----------------------|--------------------------|-------------------------------|--------------------------------|
| Wood et al. [76]            | Jamming Hole  | JAM: A Jammed-Area Mapping Service for Sensor Networks | Anomaly Based     | Uniform                  | Isotropic, homogenous | Cross-layer (MAC, routing, application) interaction | Distributed               | Location, Carrier Sense Multiple Access (CSMA) | GloMoSim                       | Jammer Range, No. of Failures |
| Raymond et al. [82]         | Jamming Hole  | RSSI and PDR based DoS Defense Mechanism | Anomaly Based     | Random                   | Isotropic, homogenous | Physical Layer | Distributed               | Received Signal Strength Indicator, Carrier Sense Time, Packet Delivery Ratio | Crossbow Mica2 and TMOTE Mini | Poor Link Identification, Jamming Detection |
| Dwivedi et al. [84]         | Jamming Hole  | Grasshopper Optimization Algorithm        | Rule-Based        | Random                   | ---            | Transport Layer | Centralized | Swarm Intelligence | MATLAB R2016a | Detection Rate, Accuracy, False Positive Rate, F-Measure, Precision, AUC (area under the curve) |
| Elie T. FUTE et al. [85]    | Black Hole    | OAO/DV Intrusion Detection System         | Hybrid             | Random                   | Isotropic, homogenous | AODV Protocol | Centralized | Intelligent Watchdog | NS-2                   | Packet Delivery Ratio, Network Throughput, End-to-End Delay |
| Sander et al. [89]          | Black Hole    | Jensen-Shannon Divergence Based Independent Component Analysis Technique | Hybrid             | Uniform                  | Isotropic, homogenous | Low-Energy Adaptive Clustering Hierarchy (LEACH) | Distributed | Energy, Trust, Cooperative Counts | NS-2.34                  | Detection Rate, Detection Time, False Alarm Rate, Packet Delivery Ratio |
| Nihiyasamad et al. [90]     | Sink Hole     | Artificial Bee Colony - Attack Detection (ABC-ADV) Algorithm | Rule-Based        | Random                   | Isotropic, heterogenous | ---            | Centralized | Swarm Intelligence | Authors Designed Testbed | Energy Consumption, Packet Loss, Packet Delivery Ratio |
| Babaei et al. [91]          | Sink Hole     | Homomorphic Encryption and Watermarking based Technique | Encryption Based  | Random                   | Isotropic, homogenous | Threshold Energy Efficient sensor Network (TEEN) protocol | Centralized | Encryption, Watermarking, Key Exchange | OMNET++                       | Throughput, Energy Consumption, Packet Delivery Ratio, Network Delay |
| Naif Abdessalem et al. [97] | Worn Hole     | Suspicious Link Detection and Worn Hole Verification in OLSR | Hybrid             | Random                   | Isotropic, heterogenous | Optimized Link State Routing (OLSR) | Distributed | Four-Way Handshaking Message | NS-2                   | Delay, Detection Rate, Accuracy |
| Aliady et al. [98]          | Worn Hole     | Energy preserving Worn Hole identification in AODV | Rule-Based        | Random                   | Isotropic, homogenous | Ad-Hoc On-Demand Distance Vector Routing | Distributed | Connectivity and Neighborhoood Information | NS-3                   | Energy Consumption, End-to-End Delay |
then nodes are declared legitimate, if not, then nodes A and B are added to the list of malicious nodes creating wormholes in the network.

These two stages are repeated for every node A and B in the path. The algorithm uses a very simplistic approach and doesn’t require any additional computational capability or hardware and requires very little energy to execute. Though the scheme is energy-efficient and cost-effective, if two malicious nodes just pretend to have the same list of neighbor nodes then it will not be able to detect the attack.

**IV. SUMMARY AND CONCLUSION**

The paper has presented a review of holes problems in WUSNs. While undergoing the entire study, the impetus has been laid on recurring holes problems and an attempt is made to have possible solutions to these problems. The effort put in would certainly be handy for the researchers in curbing the issues in WSNs.

A detailed thoughtfulness has been given to cover every issue. The work done brings into limelight various issues like coverage problems, connectivity, routing, and security. In a nutshell, it is worthwhile to mention that this paper covers minute details encompassing existing holes problems in WSNs, techniques to identify the same, and probable solutions to cope up with these holes for the enhanced efficiency in WSNs. This paper shall provide a platform to comprehend the fundamentals and put insight for further extension of the research work. A brief outline of various techniques discussed above to identify different categories of holes is summarized in the table given below Table 5-7.

**V. FUTURE DIRECTIONS**

There are many notable points and facts which are explained below:

- Most algorithms assume that the sensor network is homogeneous. Enhancement of these algorithms for heterogeneous networks can be a valuable addition.
- A very few algorithms have been developed to tackle jamming holes. Most of the algorithms work by considering traffic threshold, which may not be appropriate in all cases. The addition of jamming hole detection can be a good step in the future direction.
- In most of the algorithms, the sensing area of sensor nodes is considered as a circular disk. Algorithms can be enhanced for sensor nodes with irregular sensing areas.
- Many hole detection algorithms have poor scalability. Algorithms can be improved to show better scalability and low complexity.
- Most algorithms require location information and time synchronization and this information may not be very much accurate all the time. There is a need to analyze the effect of inaccuracies of information on the algorithms.
- Most of the techniques consider that nodes are being deployed in a 2-D environment. Future work can be done in this direction.
- The artificial intelligence-based model can be developed for security hole identification based on data collected in past.
- As it is a very cumbersome task to replace underground sensor nodes. So, more energy-efficient algorithms can be worked upon to increase network lifetime.

**ACKNOWLEDGMENT**

Priyanka Sharma is employed in the Semi-Conductor Laboratory, Department of Space, India, and is grateful to Dr. Surinder Singh, Director SCL, Dr. Manish Kumar Hudda, Head TDD/SCL and the organization for providing support for pursuing research.

**REFERENCES**

[1] I. F. Akyildiz and E. P. Stuntebeck, “Wireless underground sensor networks: Research challenges,” *Ad Hoc Netw.*, vol. 4, no. 6, pp. 669–686, Nov. 2006.

[2] M. Li and Y. Liu, “Underground structure monitoring with wireless sensor networks,” in *Proc. 6th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 69–78.

[3] M. C. Vuran and A. R. Silva, “Communication through soil in wireless underground sensor networks—theory and practice,” in *Sensor Networks*. Berlin, Germany: Springer, 2010, pp. 309–347.

[4] G. Howe, “On the transmission of electromagnetic waves,” *Electric. vol. 72.,* pp. 484–486, Jan. 1926.

[5] R. L. Van Dam, B. Borchers, and J. M. Hendrickx, “Methods for prediction of soil dielectric properties: A review,” *Proc. SPIE*, vol. 5794, pp. 188–198, Jun. 2005.

[6] J. Vasquez, V. Rodriguez, and D. Reagor, “Underground wireless communications using high-temperature superconducting receivers,” *IEEE Trans. Appl. Supercond.*, vol. 14, no. 1, pp. 46–53, Mar. 2004.

[7] N. Gupta, N. Kumar, and S. Jain, “Coverage problem in wireless sensor networks: A survey,” in *Proc. Int. Conf. Signal Process., Commun., Power Embedded Syst. (SCOPES)*, Oct. 2016, pp. 1742–1749.

[8] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. Hoboken, NJ, USA: Wiley, 2009.

[9] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. Hoboken, NJ, USA: Wiley, 2007.

[10] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, vol. 4. Hoboken, NJ, USA: Wiley, 2010.

[11] C. Guy, “Wireless sensor networks,” *Proc. SPIE*, vol. 6357, Oct. 2006, Art. no. 63571I.

[12] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*. New York, NY, USA: Springer, 2006.

[13] J. A. Stankovic, “Wireless sensor networks,” *Computer*, vol. 41, no. 10, pp. 92–95, 2008.

[14] N. Jabeur, N. Sahli, and I. M. Khan, “Survey on sensor holes: A cause-effect-solution perspective,” *Proc. Comput. Sci.*, vol. 19, pp. 1074–1080, Jan. 2013.

[15] N. Ahmed, S. S. Kanhere, and S. Jha, “The holes problem in wireless sensor networks: A survey,” *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 2, pp. 4–18, 2005.

[16] A. Sangwan and R. P. Singh, “Survey on coverage problems in wireless sensor networks,” *Wireless Pers. Commun.*, vol. 80, no. 4, pp. 1475–1500, 2015.

[17] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, “Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, Apr./May 2004, pp. 3602–3608.

[18] A. Ghosh and S. K. Das, “Coverage and connectivity issues in wireless sensor networks: A survey,” *Pervas. Mobile Comput.*, vol. 4, no. 3, pp. 303–334, 2008.

[19] C. Zhu, C. Zheng, L. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks,” *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 619–632, 2012.
[20] B. Wang, H. B. Lim, and D. Ma, “A survey of movement strategies for improving network coverage in wireless sensor networks,” *Comput. Commun.*, vol. 32, nos. 13–14, pp. 1427–1436, Aug. 2009.

[21] J. Yao, G. Zhang, J. Kanno, and R. Selmic, “Decentralized detection and patching of coverage holes in wireless sensor networks,” *Proc. SPIE*, vol. 7352, May 2009, Art. no. 73520V.

[22] G. Wang, G. Cao, and T. F. La Porta, “Movement-assisted sensor deployment,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.

[23] C. Zhang, Y. Zhang, and Y. Fang, “Localizations algorithms for coverage boundary detection in wireless sensor networks,” *Wireless Netw.*, vol. 15, no. 1, pp. 3–20, Jan. 2009.

[24] N. A. B. A. Aziz, A. W. Mohemmed, and M. Y. Alias, “A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram,” in *Proc. Int. Conf. Netw., Sens. Control*, Mar. 2009, pp. 602–607.

[25] A. Sangwan and R. P. Singh, “Coverage hole detection and healing to enhance coverage and connectivity in 3D spaces for WSNs: A mathematical analysis,” *Wireless Pers. Commun.*, vol. 96, no. 2, pp. 2863–2876, Sep. 2017.

[26] A. Boukerche and X. Fei, “A Voronoi approach for coverage protocols in wireless sensor networks,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2007, pp. 5190–5194.

[27] A. Ghosh, “Estimating coverage holes and enhancing coverage in mixed sensor networks,” in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 68–76.

[28] L.-H. Zhao, W. Liu, H. Lei, R. Zhang, and Q. Tan, “Detecting boundary nodes and coverage holes in wireless sensor networks,” *Mobile Inf. Syst.*, vol. 2016, Feb. 2016, Art. no. 8310296.

[29] C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” *Mobile Netw. Appl.*, vol. 10, no. 4, pp. 519–528, 2005.

[30] N. Yeasmin, “K-coverage problems and solutions in wireless sensor networks: A survey,” *Int. J. Comput. Appl.*, vol. 100, no. 17, pp. 1–6, Aug. 2014.

[31] S. S. Iyengar, K. G. Boroojeni, and N. Balakrishnan, “Coverage assessment and target tracking in 3D domains,” in *Mathematical Theories of Distributed Sensor Networks*. New York, NY, USA: Springer, 2014, pp. 83–108.

[32] V. De Silva, R. Ghrist, and A. Muhammad, “Blind swarms for coverage in 2-D.” in *Robotics: Science and Systems*, 2005, pp. 335–342.

[33] V. de Silva and R. Ghrist, “Coordinate-free coverage in sensor networks with controlled boundaries via homology,” *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1205–1222, Dec. 2006.

[34] A. Zomorodian and G. Carlsson, “Computing persistent homology,” *Discrete Comput. Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[35] A. Hatcher, *Algebraic Topology*. Beijing, China: Tsinghua Univ. Press, 2005.

[36] T. Kaczynski, K. Mischeakow, and M. Mrozek, *Computational Homology*, vol. 157. New York, NY, USA: Springer, 2006.

[37] Y. Bejerano, “Coverage verification without location information,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 631–643, Apr. 2012.

[38] B. Alavi and K. Pahlavan, “Modeling of the TOA-based distance measurement error using UWB indoor radio measurements,” *IEEE Commun. Lett.*, vol. 10, no. 4, pp. 275–277, Apr. 2006.

[39] C. Y. Wei, R. D. Morris, and W. A. Sethares, “Distance estimation using LEAP,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2001, pp. 1–6.
[68] B. Karp and H.-T. Kung, “GPRS: Greedy perimeter stateless routing for wireless networks,” in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[69] J. W. Jaromczyk and G. T. Toussaint, “Relative neighborhood graphs and their relatives,” *Proc. IEEE*, vol. 90, no. 9, pp. 1502–1517, Sep. 1992.

[70] R. B. Urquhart, “Some properties of the planar Euclidean relative neighbourhood graph,” *Pattern Recognit. Lett.*, vol. 1, nos. 5–6, pp. 317–322, Jul. 1983.

[71] Q. Fang, J. Gao, and L. J. Guibas, “Locating and bypassing routing holes in sensor networks,” in *Proc. IEEE INFOCOM*, vol. 4, Mar. 2004, pp. 2458–2468.

[72] Q. Fang, J. Gao, and L. J. Guibas, “Locating and bypassing holes in sensor networks,” *Mobile Netw. Appl.*, vol. 11, no. 2, pp. 187–200, Apr. 2006.

[73] P. Ghosh, J. Gao, A. Gasparri, and B. Krishnamachari, “Distributed hole detection algorithms for wireless sensor networks,” in *Proc. IEEE 11th Int’l Conf. Mobile Ad Hoc Sensor Syst.*, Oct. 2014, pp. 257–261.

[74] A. M. Khedir and W. Osamy, “Finding perimeter of query regions in heterogeneous wireless sensor networks,” *Comput. Inform.*, vol. 29, no. 5, pp. 801–821, 2012.

[75] A. M. Khedir, W. Osamy, and D. P. Agrawal, “Perimeter discovery in wireless sensor networks,” *J. Parallel Distrib. Comput.*, vol. 69, no. 11, pp. 922–929, Nov. 2009.

[76] A. D. Wood, J. A. Stankovic, and S. H. Son, “IAM: A jammed-area mapping service for sensor networks,” in *Proc. 24th IEEE Real-Time Syst. Symp.*, Dec. 2003, pp. 286–297.

[77] W. Xu, K. Ma, W. Trappe, and Y. Zhang, “Jamming sensor networks: Attack and defense strategies,” *IEEE Netw.*, vol. 20, no. 3, pp. 41–47, May/Jun. 2006.

[78] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 293–315, Sep. 2003.

[79] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Wormhole detection in wireless ad hoc networks,” Dep. Comput. Sci., Rice Univ., Houston, TX, USA, Tech. Rep. TR01-384, 2002.

[80] M. Li and B. Yang, “A survey on topology issues in wireless sensor network,” in *Proc. ICFWN*, 2006, p. 503.

[81] S. Ould Amara, R. Baghdad, and M. Oussalah, “Securing wireless sensor networks: A survey,” *EDPACS*, vol. 47, no. 2, pp. 6–29, Feb. 2013.

[82] D. R. Raymond and S. F. Midkiff, “Denial-of-service in wireless sensor networks: Attacks and defenses,” *IEEE Pervasive Comput.*, vol. 7, no. 1, pp. 74–81, Jan. 2008.

[83] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[84] S. Dwivedi, M. Vardhan, and S. Tripathi, “Defense against distributed DoS attack detection by using intelligent evolutionary algorithm,” *Int. J. Comput. Appl.*, pp. 1–11, Jan. 2020.

[85] A. Fathy, “Recent meta-heuristic grasshopper optimization algorithm for optimal reconfiguration of partially shaded PV array,” *Sol. Energy*, vol. 171, pp. 638–651, Sep. 2018.

[86] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, “Improved grasshopper optimization algorithm using opposition-based learning,” *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018.

[87] A. K. Shukla, P. Singh, and M. Vardhan, “A new hybrid wrapper TLBO and SA with SVM approach for gene expression data,” *Inf. Sci.*, vol. 503, pp. 238–254, Nov. 2019.

[88] E. T. Fute, A. T. Nangue, and E. Tonye, “An efficient and secured AODV protocol against black hole attacks on wireless sensor networks,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 18, no. 6, pp. 112–119, 2020.

[89] A. C. Sunder and A. Shannumugam, “Jensen–Shannon divergence based independent component analysis to detect and prevent black hole attacks in healthcare WSN,” *Wireless Personal Commun.*, vol. 107, no. 4, pp. 1607–1623, 2019.

[90] N. Nithyavandam and P. Latha, “Artificial bee colony based sinkhole detection in wireless sensor networks,” *J. Ambient Intell. Humanized Comput.*, vol. 7889, pp. 1–14, Jul. 2019.

[91] H. A. Babaer and S. A. Al-Ahmadhi, “Efficient and secure data transmission and sinkhole detection in a multi-clustering wireless sensor network based on homomorphic encryption and watermarking,” *IEEE Access*, vol. 8, pp. 92098–92109, 2020.

[92] A. Manjeshwar and D. P. Agrawal, “TEEN: ARouting protocol for enhanced efficiency in wireless sensor networks,” in *Proc. IPDPS*, vol. 1, 2001, p. 189.

[93] L. Xiang, J. Luo, and A. Vasilakos, “Compressed data aggregation for energy efficient wireless sensor networks,” in *Proc. 8th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2011, pp. 46–54.
MAZIN ABED MOHAMMED received the B.Sc. degree in computer science from the University of Anbar, Iraq, in 2008, the M.Sc. degree in information technology from UNITEN, Malaysia, in 2011, and the Ph.D. degree in information technology from UTeM, Malaysia, in 2019. He is currently a Lecturer with the College of Computer Science and Information Technology, University of Anbar. His research interests include artificial intelligence, biomedical computing, and optimization.

RACHNA SHAH received the B.E. degree in computer science and engineering from the Kumaon Engineering College, Dwarahat, India (a Government of Uttarakhand Institution). She is currently pursuing the Ph.D. degree with IIT Guwahati, India. She is also working as a Scientist with the National Informatics Centre, Government of India. She has more than ten years of experience in software development, architecture planning and development, and e-governance. She has published numerous research articles in national and international journals, including IEEE, Springer, Taylor and Francis, Wiley, and Elsevier. Her research interests include data science, the IoT, and open-source software development. She has received various excellence awards from the Government of India and the Government of Uttarakhand.

JAN NEDOMA (Senior Member, IEEE) was born in Czech Republic, in 1988. He received the master's degree in information and communication technology from the Technical University of Ostrava, in 2014. Since 2014, he has been working with the Technical University of Ostrava as a Research Fellow. In 2018, he successfully defended his dissertation thesis and he started working as an Assistant Professor with the Technical University of Ostrava. He has become an Associate Professor in communication technologies, in 2021, after defending the habilitation thesis entitled Fiber-Optic Sensors in Biomedicine: Monitoring of Vital Functions of the Human Body in Magnetic Resonance (MR) Environment, and become the Head of the Optoelectronics Laboratory. He has more than 150 journal articles and conference papers in his research areas and nine valid patents, such as IEEE REVIEWS IN BIOMEDICAL ENGINEERING, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, IEEE ACCESS, Human-Centric Computing and Information Sciences, and Frontiers in Physiology. His research interests include optical communications, optical atmospheric communications, optoelectronics, optical measurements, measurements in telecommunication technology, fiber-optic sensory systems, biomedical engineering, data processing from fiber-optic sensors, the use of fiber-optic sensors within the SMART technological concepts (smart home, smart home care, intelligent building, smart grids, smart metering, and smart cities), and for the needs of Industry 4.0.