The Journal of Supercomputing (2023) 79:20845–20877
https://doi.org/10.1007/s11227-023-05494-8

# Node position estimation based on optimal clustering and detection of coverage hole in wireless sensor networks using hybrid deep reinforcement learning

Rajib Chowdhuri[1] · Mrinal Kanti Deb Barma[1]

Accepted: 8 June 2023 / Published online: 17 June 2023


**Abstract**

Sensor nodes, typically small and low-power devices, are components of wireless sensor networks (WSNs). Each node monitors its surroundings for relevant environmental changes and sends all detected events to the data collector for analysis. If the sensor nodes are not placed correctly, there may be areas that are not within the detection zone of any sensor node. Coverage holes in WSNs are usually caused by random deployment and node failure. Energy holes and dead nodes are the main problems caused by detection and recovery of coverage holes in WSNs. The size of coverage holes increases the time complexity and power of recent protocols. However, there is a high computational complexity associated with distributed methods proposed in recent years to solve the coverage hole detection problem. In this paper, we propose optimal cluster-based node position estimation and coverage hole detection in WSNs using a hybrid deep learning approach. First, a modified Lyapunov optimization (MLO) algorithm to compute the node position is presented, which ensures edge nodes in the network. Next, we design optimal clustering technique by using improved sand cat swarm optimization (ISCSO) algorithm to formulate efficient balanced clusters which computes coverage hole area in the network. Afterward, we develop a hybrid deep reinforcement learning (Hyb-DRL) technique for hole shape detection and hole size judgment within clusters, among clusters and along edges. The results show that the proposed approach achieves significant improvements compared to existing benchmark approaches. Specifically, the average energy consumption of CG-DCHD approach is 43.835%, 32.674% and 26.164% lower for node density, hole density and simulation rounds, respectively. The hole detection time is 18.4%, 16.802% and 15.462% lower, while the coverage is 16.885%, 14.977% and 12.219% higher for node density, hole density and simulation rounds, respectively. Additionally, the network lifetime of CG-DCHD approach is 15.58%, 17.702% and 20.492% higher, while the control packet overhead is 0.83%, 1.907% and 1.466% lower for node density, hole density and simulation rounds, respectively.

Extended author information available on the last page of the article

**Keywords** Node position · Coverage hole · Wireless sensor network · Clustering · Hole shape detection

**Abbreviations**

| | |
|---|---|
| WSN | Wireless sensor network |
| BS | Base station |
| MLO | Modified Lyapunov optimization |
| ISCO | Improved sand cat swarm optimization |
| Hyb-DRL | Hybrid deep reinforcement learning |
| QoS | Quality of service |
| Rs | Discovery range |
| Rc | Correspondence range |
| $E_{\text{total}}$ | Total energy consumption |
| RSS | Received signal strength |
| M | Mobility |

## 1 Introduction

Currently, the progress in micro-electromechanical systems (MEMS) technology has made it possible to develop compact sensor arrays that can detect, analyze and collect data from the environment in a more efficient and effective manner for remote communication. Wireless networks are referred to as wireless sensor networks (WSNs) [1] that do not require any kind of infrastructure and are made to share data while simultaneously monitoring physical or environmental conditions like temperature, noise, vibration, pressure, motion or pollution. Through the network, data can be monitored and analyzed at a primary location or sink [2]. A base station or sink serves as an interface between users and the network. By sending queries and collecting responses from the receiver, the necessary information is obtained from the network. There are typically hundreds of thousands of sensor nodes in a wireless sensor network. Due to a number of limitations, WSN enables new applications and necessitates unconventional protocol design paradigms [3]. The field has made significant efforts in research, standardization and industrial investment over the past decade. The majority of WSN research currently focuses on developing protocols and algorithms that are both energy and computationally efficient, and the scope of application is limited to simple monitoring and control applications. Although sensors are powered by batteries, they have limited processing speed and memory to detect or transmit data. As sensors are installed closer to the observed event, the probability of obtaining a large amount of accurately detected data increases [4]. WSNs have been increasingly used in a variety of security and surveillance-related applications ever since their inception tremendously. WSN, a collection of multi-resource strain sensor nodes, are broadly utilized in fields like ecological horticulture, far off tolerant observing, military reconnaissance, catastrophe forecast, checking and processing plant mechanization. The essential capability of sensor hubs in such applications is to assemble important natural information and send it to a base station (BS) or sink via multi-cast. This topological disadvantage causes neighboring nodes to drain energy significantly faster, which can create energy holes in the network [5]. WSN consist of multiple sensor nodes that send environmental data to a base station (BS) and collect it.

One of the most significant issues with WSNs is the need to maintain coverage. The batteries serve as the power source for the sensor nodes die when the battery is depleted; when multiple nodes fail, communication is lost and network coverage is interrupted [6]. In WSNs, coverage is ensured and maintained in two steps, with coverage hole detection carried out by a straightforward distributed system with low overhead. Nodes check to see if there are any holes around. At the point when the quantity of openings surpasses the, the openings with the most brief distance to the portable hubs and highest priority are chosen number of mobile nodes. Across the past decade, rapid advances in wireless communication and embedded micro-sensing technologies have led to smaller and cheaper wireless assemblies. Various nodal features related to existing work morphology and geometric distribution of connectivity are used to find gaps in coverage. However, the majority of algorithms call for a relatively high average node count degree, which results in high communication costs when searching for specific patterns [7]. It should be noted that no algorithm offers a technique for clustering nodes, despite the fact that all algorithms locate nodes near coverage gaps. Unfortunately, large communication overhead significantly shortens the life cycle of WSNs.

WSNs are used in advanced applications to detect hostile or unreachable locations such as environmental monitoring, rescue operations, pollution detection, outdoor surveillance, warfare, health care and home entertainment [8]. In addition, each node can collect, store, process and communicate information about the environment with neighboring nodes. In the monitoring area, see a coverage hole that is not covered by any sensor detection disk [9]. The objective is to provide effective coverage while minimizing energy consumption with the scalable firefly algorithm and mobile nodes. It involves evenly dividing the network area but smaller cells to facilitate physical coverage estimation [10]. To fill the gaps, a general research framework of reinforcement learning was hybridized with the branch and cut algorithm for network operators [11] which ensured the full coverage hole. The network operators need methods to combine and re-optimize antenna beams to improve network coverage during interference mitigation [12]. WSN hole detection (WHD) algorithm was developed to estimate the area of holes in ROIs where sensor nodes are distributed at random and to detect them. Based on energy consumption and designed to achieve quality of service (QoS), WHD average hole detection time [13]. Credible information coverage hole detection (CICHD) problem is solved and investigated based on credible information coverage (CIC) model [14]. WSNs can strictly monitor by FoI and whether WSNs can collect and transmit the necessary information are two important problems to be solved in WSN [15]. The sensor deployment schemes are blind spot centroid scheme (BCPS) and confused centroid scheme (TCPS) was used to solve the coverage problem in MWSNs [16].

**Our contributions** For further enhancement, we propose an optimal cluster-based node position estimation as well as WSN coverage hole detection. The following is a discussion of the major contributions that this work relies on.

1. A modified Lyapunov optimization (MLO) algorithm is used for node position computation which ensures the detection of edge nodes in the network.
2. An optimal clustering technique is perform by using improved sand cat swarm optimization (ISCSO) algorithm to formulate efficient balanced clusters which computes coverage hole area in the network.
3. A hybrid deep reinforcement learning (Hyb-DRL) technique is further used for hole shape detection and hole size judgment within clusters, among clusters and along edges.
4. Furthermore, performance of the proposed approach is evaluated by implementing the proposed technique with the NS2 simulation tool, in which node density, node mobility and node sensing range are simulated.

The following is how the remainder of this paper is organized. Section 2 discusses the WSN coverage hole detection literature review. Section 3 covers our proposed work's problem methodology and network model. The appropriate mathematical model is then used to provide a comprehensive breakdown of the proposed work's workflow Sect. 4, in addition to the simulation outcomes and a comparison of the proposed and existing coverage hole detection methods. Section 6 shows the final section in the paper.

## 2 Related works

In this section, we will look at different methods for detecting coverage holes utilized for WSN environment. The summary of research gaps is listed in Table 1.

Wang et al. [17] have proposed a general research framework for problems with coverage gaps that have unknown characteristics to summarize related models like deployment models, detection models and network models using some fundamental ideas for coverage issues involving properties that are not clear. Improve the quality of coverage, extend the life of the network, and reduce sensor count when coverage issues are uncertain characteristics are the three goals that can be accomplished using these models. Implementation, planning or selection, movement or adjustment, and the various decision strategies are all implemented. Then, divide the solutions into traditional, heuristic, approximate, distributed, centralized and stochastic algorithms according to their various aspects. Phoemphon et al. [18] have proposed using enhanced particle swarm optimization (NS-IPSO) to segment sensor nodes in order to improve the precision with which the distances between unknown nodes and pairs of anchor nodes can be estimated to find potential sensor nodes that could be used to divide up the region's anchor nodes. These sensor nodes occur more frequently than the standard deviation of all sensor nodes because they have a shorter path between anchor nodes.

**Table 1** Summary of research gaps

| Ref | Methodology | Technique used | Findings | Research gaps |
|---|---|---|---|---|
| [17] | Coverage problem with uncertain properties | Voronoi graph | Coverage quality, network lifetime | Because the probability of an event occurring varies too much, full coverage quality appears to be too strong |
| [18] | Hybrid localization model for WSN | NS-IPSO | Average location error, network lifetime | Coverage detection is inefficient due to high computational complexity |
| [19] | Coverage hole detection in WSN | Dynamic alliance network with PSO | Wake-up time and target capture rate | Not suitable to detect triangular holes |
| [20] | Polynomial mapping for node localization | Gaussian and polynomial kernel embedding | RMSE error, hole detection accuracy | A network with coverage holes will not be able to satisfactorily carry out its monitoring task |
| [21] | Distributed self-healing coverage hole detection | Fully distributed hole detection and repair | Coverage, connectivity and energy consumption | Failure in multiple hole detection |
| [22] | Malevolent node detection in WSN | Rabin–Karp | Packet forward rate, delay, throughput | Not scalable for high node density |
| [23] | Energy optimized and QoS concerned data gathering protocol | VD-PSO | Network lifetime, energy consumption, delay, packet delivery ratio | Leads to higher energy consumption incurred by sensor nodes movements |
| [24] | Computational geometry-based coverage hole detection | Delaunay Triangulation | Coverage, connectivity and energy consumption | Require relatively high average node degrees which leads high communication overhead |
| [25] | Location privacy and congestion avoidance | Jellyfish dynamic routing protocol (JDRP) | Energy consumption, packet delivery delay and lifetime | Identify coverage holes by locating the nodes on the boundary of holes, none of them provides details regarding clustering |

Jia et al. [19] have proposed in order to monitor partial discharges of high voltage (HV) equipment, a monitoring network was created. A strategy for installing ultrasonic sensors to monitor partial discharges is utilized in accordance with the three-dimensional characteristics of the devices. An adaptive angle adjustment-based virtual force algorithm is proposed. It has been demonstrated that applying this algorithm covers more than 94% of the target within a manageable integration range. A sensor collaboration scheme is proposed based on a standard security protocol, which verifies its energy saving and delay compression effects and guarantees the target acquisition rate. Xu et al. [20] have proposed the Gaussian kernel integration algorithm, the polynomial kernel integration algorithm and the polynomial graph-based semi-supervised manifold learning algorithm A low-dimensional physical space and a high-dimensional location data space can be linked with a polynomial mapping function be calculated using this method, which has a high discrimination rate and clear nonlinear feature mapping. The performance of this algorithm is then evaluated and compared to some related localization techniques under various signal noises, anchor nodes and communication ranges, respectively. Khalifa et al. [21] have proposed a distributed self-healing algorithm known as distributed hole detection and repair (DHDR) makes use of just existing network nodes to simultaneously find and fix holes. It accurately estimates the size and location of a cover hole is dynamically detected as it occurs. By exchanging data and coordinating their movements, it selects suitable nearby nodes that maximize coverage while consuming the least amount of energy. The selected nozzles move in such a way that they restore the hole's hollow area without affecting the connection or coverage that is already in place. Vishnupriya et al. [22] have developed the Rabin–Karp algorithm to separate the malicious sensor from the network. It validates the sensor's authenticity within the WSN and eliminates the eavesdropping attack. The cluster head (CH), who is chosen based on cooperation rank, data transmission rank and residual energy, receives maintenance data from sensor nodes. CH detection data is gathered with the help of a mobile node via intermediate channels. Thus, it reduces latency and usage of excess energy in the network. Roy et al. [23] have investigated the issues related to quality of service (QoS) and a proposed energy-proficient WSN versatile information gatherer convention for information assortment Numerous asset requirements of WSN hubs make it truly challenging to protect nature of administration, particularly because of energy restriction. Das et al. [24] have proposed a computational geometry-based and empty circle-based framework for estimating coverage holes and hole areas. The Delaunay triangle is constructed using the location information of sensor nodes to identify a coverage gap in a large-scale WSN. After locating the hole, the next objective is to estimate its area. The empty circle property can be used to determine whether or not a coverage hole exists in a specific ROI of a large-scale WSN. The coverage hole detection and area estimation (CHDAE) efficiency of the algorithm is also demonstrated through simulations. Christopher et al. [25] have come up with a low-latency Jellyfish dynamic routing protocol (JDRP) to avoid congestion and safeguard location privacy. Here, the transmission distance is calculated for each subsection, which selects the target area from the entire sensor field.

## 3 Problem methodology and network model

### 3.1 Problem methodology

Ma et al. [26] have proposed a computational geometry-based, distributed protocol for remotely detecting coverage holes in self-organized WSNs. Calculations rather than predictions regarding the presence of holes at the site are used to detect holes in the current card. Geometric sampling of just one or two neighbors from each node is used in the process of hole detection, which is straightforward but efficient. It is suggested that local node data serve as the foundation for a global coverage hole detection strategy. Each node's communication range is assumed to be equal to its detection range in contrast to other coverage hole detection protocols, allowing for greater energy savings during communication. After the deployment of the network, coverage gaps compromise this task. Coverage gaps are inevitable for a variety of reasons, including physical damage and depletion of sensor node energy. As a result, it is critical to have continuous coverage monitoring system as coverage holes can negatively affect network performance if left unattended. In general, coverage is considered as an important quality of service (QoS) measure to increase monitoring quality in the target area. Holes in WSNs can occur due to sensor placement issues or power/hardware failure. In such cases, the detection or data transmission may affect the WSN's normal operation. It can also reduce sensor detection coverage and network lifetime. Due to the fact that the precise location of sensors is frequently unknown, the hole detection problem in WSNs is particularly challenging.

Existing research [17–26] introduces a new node that joins the network or active nodes that are already in use to recover from coverage loss owing to the complications of conventional redundant hole detection methods such as extra time, low accuracy and high operating cost [19, 20]. Adding new nodes to the network when a coverage gap occurs not possible if the region of interest is in an unfavorable location [23]. Changing or changing the detection range of active nodes not only changes the network topology constantly introduces new coverage gaps and increases coverage overhead [22]. Maintaining network coverage, energy wireless sensor networks face significant difficulties in terms of lifetime and power consumption [17–20]. Since the sensor node battery cannot be replaced or recharged, the battery discharge lasts for the lifetime of the sensor node. The network coverage is reduced when some sensor nodes fail and disconnect decreases compromised [26]. Specifically, when a sensor node fails the main objective of the coverage and communication optimization model is to select some sensor nodes with maximum direct sensor node connectivity [25]. However, the existing methods do not allow obtaining a minimum selection of nodes, thereby alleviating the bottlenecks of traditional coverage methods. Hole detection is performed by taking one or two neighboring hops from each node using simple but effective geometric methods [26]. However, this method is not suitable for multi-hop routing and cannot be detected. To our knowledge, very few algorithms have suggested detecting coverage holes. Additionally, the majority of these protocols take into account a typical monitoring area, rather than the actual sensor location randomly aligned, it is more convenient to obtain a region of random arrangement. To overcome above problems by optimal cluster-based node position estimation and coverage hole detection approach using hybrid deep learning technique. The major objectives of our proposed approach are summarized as follows:

1. The joint optimization of node position estimation and coverage hole detection approach is used to avoid random deployment and node failure.
2. To design optimization algorithm to compute the exact location of sensor nodes in the network this ensures the edge nodes
3. To introduce clustering algorithm to detect the coverage holes in the network and formulate objective function
4. The hybrid deep learning technique finalizes the hole shape and hole size judgment for further decision-making which improves detection accuracy

### 3.2 Network model

We assume that n sensor nodes are designed to detect particular events and are arranged at random R in a rectangular field. Within its transmission range, every sensor hub can speak with different hubs and detect certain events. The sensor hubs are sufficiently thick and there are no disconnected sensor hubs. Sensor hubs realize their area utilizing GPS or other following strategies.

We also assume that every sensor hub realizes point by point data including the IDs and directions of its communication neighbors. This information can be collected through a beacon, a hello message or a series of topological changes expected. During network training, every hub learns its area data and gathers a rundown of one-bounce and two-jump neighbors. Figure 1 shows the organization model of our proposed way to deal with explore a multi-bounce remote organization with hubs with various handsets. The hubs of the objective locale are haphazardly disseminated inward hubs, and different hubs are consistently appropriated along the external limit of the objective district to guarantee total inclusion. Every hub does not know explicit area data, so a hub might be set as the default interior hub. The discovery range (Rs) is equivalent to the correspondence range (Rc), and every hub sends its area data through GPS or some other area data framework. In this organization model, regardless of whether a piece of the organization has an inclusion opening, the whole organization is completely associated. In any case, it is accepted that no separated sensor shows up in the organization on the grounds that the whole organization is completely associated, and the spotted circle between the two demonstrates twofold series correspondence.

## 4 Proposed methodology

In this segment, we portrays the functioning system of proposed optimal cluster-based node position estimation and coverage hole detection approach, which consists three fold process are node position estimation, coverage hole detection, hole shape detection and hole size judgment.

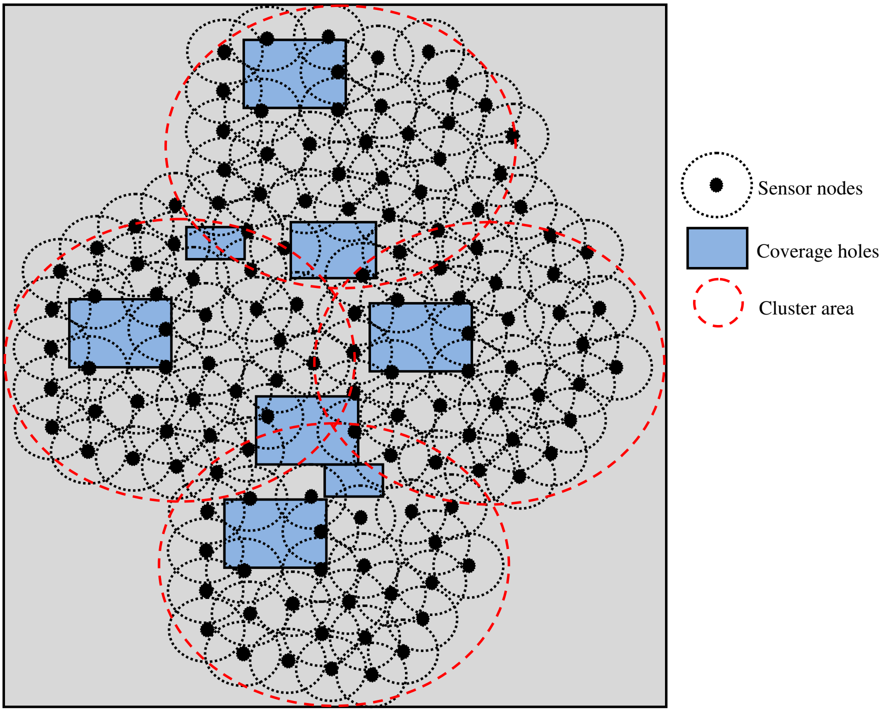


**Fig. 1** Network model of proposed approach

### 4.1 Node position estimation

In this stage, every hub gathers and decides the vital data about its current circumstance, which is utilized to set up the following stage, track down undesirable neighbors and lastly decide the convergence focuses with the opening. Most existing location algorithms can be classified as threshold-based or threshold-based, depending on whether the algorithm uses distance estimation or other information to calculate node locations. Therefore, it is necessary to develop new techniques, methods and methods to solve the localization and localization problem of wireless sensor nodes. In this work, the sensor node position estimation is depends on three important design constraints are received signal strength, interference range and distance between sensor nodes, sink node. Here, we noted that all the design metrics are considered as the time varying factors, so need to optimize it. For that reason, a modified Lyapunov optimization (MLO) algorithm is used and estimates the exact node position and ensures the detection of edge nodes in the network.

#### 4.1.1 Design constraints for node position estimation

Received signal strength (RSS) is commonly used standard because it is easy to measure and directly related to service quality. RSS feeds and mobile terminals are closely related to its connection point. The power consumption of transmit and receive domains is derived from the basic output power model taking into account the power demand. The energy consumption of sensor nodes depends on the amount of data and the transmission spacing. The vigor ingestion of a node ($n$) is comparative with square of distance when the expansion distances ($I$) not exactly the starting distance. The total energy consumption of each sensor node is compute as follows.

$$E_{\text{total}} = S\,(n,\,I) + R(n) \tag{1}$$

where $S\,(n,\,I)$ and $R\,(n)$ are energy ingestion of transmission and acquire node.

$$S(n, I) = \begin{cases} n \times E_{\text{elec}} + n \times \varepsilon_{\text{fs}} \times I^2; & \text{if} \quad I < I_0 \\ n \times E_{\text{elec}} + n \times \varepsilon_{\text{mp}} \times I^4; & \text{if} \quad I \geq I_0 \end{cases} \tag{2}$$

$$R\,(n) = n \times E_{\text{elec}} \tag{3}$$

The power required to operate the transmitter or receiver circuit per unit area, denoted by $S(n,I)$, determines the power consumption for both the escaped space and multi-path models, which is influenced by the source communication model and initial communication distance. It is important to note that all costs considered in this study are benign for energy consumption. The RSS metric is independent of distance and communication energy. When the node transmits packets with power, the received signal strength (RSS) is determined by the distance "$I$," and is calculated as follows:

$$\text{RSS} = \frac{S(n, I)}{4\pi\, i_i^2} + s_{a,\, a_1/a_2} \tag{4}$$

The sign strength of the ongoing example not entirely set in stone by the development, distance and relative speed, and the sample points are selected and controlled $\Delta s_1 = \Delta s_2 = \Delta s$, but such points are not present in the sample field. Different reference points are used for the signal strength actually received from the potential nodes, where $I_{i_1}$, $I_{i_2}$ and $I_{i_3}$ can be acquired and adapted distance is figured from the cosines regulations (A) as follows:

$$I_{i_1}^2 = I_{i_2}^2 + A_1A_2^2 - 2I_{i_2} \cdot A_1A_2 \cdot \cos\,(\alpha) \tag{5}$$

$$I_{i_3}^2 = I_{i_3}^2 + A_1A_2^2 - 2I_{i_2} \cdot I_1I_2 \cdot \cos\,(\beta) \tag{6}$$

The speed and distance of the sensor nodes are determining factors for reaching the target position ($V$), which describes as follows.

$$2A_1A_2^2 = I_{i_1}^2 + I_{i_2}^2 - 2I_{i_3}^2 \tag{7}$$

$$V = \sqrt{\frac{2\left(I_{i_1}^2 + I_{i_2}^2 - 2I_{i_3}^2\right)}{2\Delta s}} \tag{8}$$

The motion continuance $S_{a,a_1/a_2}$ for sensor node from genuine spot to the impacted position or is expressed as the distance separated the hub's speed and it can secure by sign regulation as follows:

$$S_{a,a_1/a_2} = \frac{R \cdot \sin \vartheta}{\sin \beta \cdot V} \tag{9}$$

Finally, we get the following mobility (M) function as,

$$M = S_{a,a_1/a_2} = \frac{\Delta s \cdot R \cdot \sin \vartheta}{\sin \beta \cdot \sqrt{\frac{\left(I_{i_1}^2 + I_{i_2}^2 - 2I_{i_3}^2\right)}{2}}} \tag{10}$$

Interference range is compute from the sensor node coverage range. Assume there are two sensors (0, 0, 0) and (1, 0, 0). The awareness scope of the two sensors is characterized as follows.

$$y^2 + x^2 + z^2 = r_i^2 \tag{11}$$

$$(Y - Y_{ji})^2 + X^2 + Z^2 = R_i^2 \tag{12}$$

The distance between where the two sensor congregations structure a three-layer focal point is equipped for estimating the state of the two circular covers.

$$G_j = \frac{(R_j - R_i + Y_{ji})(R_j - R_i + Y_{ji})}{2Y_{ji}} \tag{13}$$

$$G_i = \frac{(R_i - R_j + Y_{ji})(R_j + R_i - Y_{ji})}{2Y_{ji}} \tag{14}$$

The congestion rate is underneath and is between them.

$$v_{\text{overlap}} = v(R_j, G_j) + v(R_i, G_i) \tag{15}$$

The updated solution is compute as follows:

$$v(R, G) = \frac{\pi}{3} G^2 (3R - G) \tag{16}$$

$$v_{\text{overlap}} = \frac{\pi}{12Y_{ji}} (R_j - R_i - Y_{ji})^2 (Y_{ji}^2 + 2Y_{ji}R_i - 3R_i^2 + 2Y_{ji}R_j + 6R_jR_i - 3R_i^2) \tag{17}$$

In the event that the sensor span is no different for all sensors because of balance and change the awareness range as ($R_j = R_i = R$)

$$v_{\text{overlap}} = \frac{\pi}{12}(2R - Y_{ji})^2(Y_{ji} + 4R) \tag{18}$$

The typical relationship coefficient is communicated as follows:

$$\rho(j,\, i) = \frac{v_j^i + v_i^j}{v} \tag{19}$$

where $v_{\text{overlap}} = v_j^i + v_i^j$ and update solution as follows

$$\rho(J,\, I) = \frac{1}{16R^3}(\text{cor} - y_{ji})^2(y_{ji} + 2 \times \text{cor}) \tag{20}$$

where $\text{cor} = 2R$ is the switch boundary.

$$\rho(j,\, i) = \left\{ \begin{array}{ll} \frac{1}{16R^3}(\text{cor} - y_{ji})^2(y_{ji} + 2 \times \text{cor}), & \text{if} \quad 0 \le y_{ji} < \text{cor} \\ 0, & \text{if} \quad y_{ji} \ge \text{cor} \end{array} \right\} \tag{21}$$

The distance among sensors and sink hub is figure by utilizing the conveyance proportion calculation. The closeness of the connecting terminal is displayed as follows:

$$n_q = \frac{1}{n_{\text{sos}}} \sum_{j=1}^{n_{\text{sos}}-1} \text{dist}(n, j) \tag{22}$$

where which gauges the distance among *Tn* and its neighbor *Tn* is without a doubt the quantity of touching *Tn*. Now, we applied modified Lyapunov optimization (MLO) algorithm for node position computation which ensures the detection of edge nodes in the network.

Lyapunov enhancement algorithm allows to the utilization of Lyapunov capability to control a unique framework ideally. Lyapunov capabilities are generally utilized in charge hypothesis to guarantee the stability of various types of systems. A multidimensional vector often describes the state of a system at a particular moment in time. In contrast, the MLO algorithm is inspired by the basic Lyapunov function, which is a nonnegative proportion of this multifaceted state. As a general rule, activity develops when the system changes to undesirable conditions. The stability of the system is obtained by performing control measurements, which cause the Lyapunov function to deviate in the negative direction toward zero. Here, Lyapunov capability of the line as. The persistent time Lyapunov float generator is characterized as,

$$\Delta(P) = \lim_{\delta \to 0} \frac{E[v(P)(S + \delta)) - V(P(S))|P(S)]}{\delta} \tag{23}$$

By the going with lemma, the Lyapunov float $\Delta(Q)$ can similarly be gotten from the going with SDE. Following from the Lyapunov smoothing out framework, we add the discipline term to obtain the float notwithstanding discipline term, i.e., where $v > 0$ is control limit to control the power–defer trade-off. Then, we have the accompanying arrangement in regard to the float in addition to punishment term.

$$\Delta v(P) \leq \Theta + VE[q_{\mathrm{sos}}|P] - m = \sum^{m} P_m E[U_M - \mathbf{B}_M|P] \tag{24}$$

To apply Lyapunov advancement hypothesis, we initially change the drawn out typical imperatives into virtual sequences. Two virtual arrays $A(s)$ and $Q(s)$ can be defined below the battery and BER threshold, respectively.

$$A(s+1) = \max\{A(s) + \theta - a(s+1), 0\} \tag{25}$$

$$Q(s+1) = \max\{Q(s) + \phi(s) - \delta,\ 0\} \tag{26}$$

Virtual queues can be considered as signals to determine whether constraints have been encountered in previous time slots. At higher $A(s)$, battery voltage is sacrificed due to earlier continuous transfer. Also, at the network level, each BS maintains a set of internal queues to store the current backlog of its users. Let $Q_{NL}(s)$ represents the current backlog of $l$th user in nth BS. Then the evolution of the size of $Q_{NL}(s)$ is given by

$$Q_{NL}(s+1) = \max[Q_{NL}(s) + R_{NL}(s),\ 0] + b_{NL}(s), \tag{27}$$

For all $N \in n$ and $L \in l(N)$, where is the transmission rate offered to the $i$th user of the nth BS in the time slot, we adopt the concept of strong stability, and the network is very stable.

$$\overline{P}_{NL} = \lim_{s\to\infty} \sup \frac{1}{s} \sum_{\tau=1}^{s} e\{P_{NL}(\tau)\} < \infty, \quad N \in n,\ L \in l(N) \tag{28}$$

There the waiting depends on the control policy, which is related to the random state of the channel and the control actions taken in response to those channel states. Intuitively, this expression means that a sequence is strongly stationary if its time mean is finite. The network is very stable if all individual sequences of the network are highly stable. Let $Y_N(s)N \in n$ be virtual queues associated with constraint. We update the virtual queue $Y_N(s)$ for all $N \in n$ at each time slot as

$$Y_N(s+1) = \mathrm{Max}[Y_N(s) - y_N^{\mathrm{out}}(s),\ 0] + y_N^{\mathrm{in}}(s) \tag{29}$$

Likewise, to ensure inequality constraint and define virtual queues $X_N(s)N \in n$; and update $X_N(s)$ for all $n \in N$ according to the following dynamics

$$Y_N(s+1) = \mathrm{Max}[X_N(s) - x_N^{\mathrm{out}}(s),\ 0] + x_N^{\mathrm{in}}(s) \tag{30}$$

Then, we define a quadratic Lyapunov function $l(\Theta(s))$ as,

$$l(\Theta(s)) = \frac{1}{2}\left[\sum_{N\in n}\sum_{L\in l(n)} P_{NL}(s)^2 + \sum_{N\in n} Y_N(s)^2 + \sum_{L\in l(n)} X_N(s)^2\right] \tag{31}$$

The Lyapunov function is a metric that measures network congestion. Intuitively, if is short, then all lines are short. If larger, then at least one row is larger. The drift of the Lyapunov function (i.e., the expected change from one point of the Lyapunov function to another) can be written as.

$$\Delta(\Theta(s)) = e\{l(\Theta(s+1)) - l(\Theta(s))|\Theta(s)\} \tag{32}$$

We use the introduced drift plus penalty minimization method to solve the problems. This control principle solves the problem by reducing the bottom drift and the upper bound on the penalty exposure.

$$\Delta(\Theta(s)) = -v \sum_{N \in n} e\{\log(\mu_N(s))|\Theta(s)\} \tag{33}$$

where $V \geq 0$, subject to the constraint in each time slot. The working process of our proposed node position estimation using MLO is described in Algorithm 1.

**Algorithm 1 Node position estimation using MLO**

Input : define design constraints from sensor nodes
Output : node position

1. Generate a random parameter
2. Define continuous-time Lyapunov drift generator
$$\Delta(P) = \lim_{\delta \to 0} \frac{E[v(P)(S+\delta)) - V(P(S)) \mid P(S)]}{\delta}$$
3. While **Do**
4. For i=1and j=0
5. The evolution of the size of $Q_{NL}(s)$ is given by
$$Q_{NL}(s+1) = \max[Q_{NL}(s) + R_{NL}(s), 0] + b_{NL}(s),$$
6 Update the virtual queue $Y_N(s)$ for all N ∈ n at each time slot as
$$Y_N(s+1) = Max[X_N(s) - x_N^{out}(s),\ 0] + x_N^{in}(s)$$
7. The drift-plus-penalty expression
$$\Delta(\Theta(s)) = -v \sum_{N \in n} e\{\log(\mu_N(s)) \mid \Theta(s)\}$$
8. Compute the final fitness value
9. End for
10. End

## 4.2 Cluster-based coverage hole area detection

Here, we utilized optimization techniques to identify areas with no coverage by the deployed sensors. In this approach, the network is divided into clusters of nodes, and the coverage hole area within each cluster is computed. The goal is to detect and localize coverage holes as accurately as possible, which is important for maintaining network connectivity and optimizing the use of resources. The cluster-based approach is preferred over a centralized approach because it reduces the amount of data that needs to be transmitted to a central location, which in turn reduces the energy consumption of the nodes. Moreover, the distributed approach is more robust to node failures or network partitions, as each cluster can operate independently. The cluster-based coverage hole area detection technique involves several steps, including cluster formation, coverage hole detection and hole area computation.

Improved sand cat swarm optimization (ISCSO) is a metaheuristic optimization algorithm that is inspired by the hunting behavior of sand cats. ISCSO is used in this paper for cluster-based coverage hole area detection in wireless sensor networks. The goal of this algorithm is to generate optimal cluster heads with maximum coverage and minimum overlap. ISCSO uses a set of sand cat individuals, which move in search of the optimal solution. The algorithm initializes a population of sand cat individuals, and each individual represents a potential cluster head. These individuals move in the search space, which is defined by the network area. Each individual evaluates the fitness of the solution, which is measured by the coverage area and the overlap with other clusters. The algorithm uses a combination of exploration and exploitation strategies to find the optimal solution. The exploration strategy is based on the movement of sand cat individuals in the search space, while the exploitation strategy is based on the selection of the best individuals to generate new solutions. ISCSO has shown promising results in solving optimization problems, and it has been applied in various fields such as engineering, finance and image processing. In this paper, ISCSO is used to generate optimal clusters for coverage hole detection in wireless sensor networks, which improves the overall performance of the system. In the early stages of breeding, groups of cats can achieve strong universal optimization capabilities.

$$R_K = \frac{1 - (\sqrt{C_K + 1}) \cdot \ \text{rand} \cdot j}{C_K \cdot \text{MaxIter}}, \quad K = 1, 2, j < \frac{\text{MaxIter}}{2} \tag{34}$$

The equations can be used to change the ability to search globally in the early stages of particle moving and to improve local refinement and resolution accuracy in subsequent iterations. Added a radius limits to the sensor nodes search location and position. When the distance $y_j^D$ and $h_{\text{best}}^D$ radius are less than the individual $y_j^D$ to $h_{\text{best}}^D$. However, when the distance $y_j^D$ and $q_{\text{best}}^D$ radius between them is small, deviate from $q_{\text{best}}^D$. The value of the elements is determined by the following equation:

$$\begin{aligned} V_j^D(s+1) = \omega \cdot V_j^D(s) + C_1 \cdot R_1 \cdot (q_{\text{best}}^D(s) - y_j^D(s)) \\ + C_2 \cdot R_2 \cdot (h_{\text{best}}^D(s) - y_j^D(s)) + F \cdot f_j + E \cdot e_j, \end{aligned} \tag{35}$$

If the underlying upsides of and are moderately little, add them to control the arrangement of negative qualities to stay away from negative numbers.

$$R_1 = R_2 = \text{rand}, \quad R_1 \leq 0, \ R_2 \geq 0. \tag{36}$$

Particles study each other to get the most enlightening data in their fields. The memory component of $x$ is added to each finder, giving each particle a smaller memory load than before. Specifies more memory weights are used to improve the current level and historically optimal level of each query.

$$y_j^D(s+1) = \frac{1}{2}(x \cdot y_j^D(s) - (1-x) \cdot y_j^D(s-1) + x \cdot V_j^D(s+1) + (1-x) \cdot V_j^D(s)) \tag{37}$$

First, start with each cat approximately and compute the cost of the exercise. Finally, adjust its settings in tracing mode. After adding a section, we propose an advanced transfer learning model that applies three improved strategies to this small sketch data set. It consists of two parts: sample selection and sample refinement. The second term G is constructed so as not to contribute to the BCs, since $x_s(y)$ satisfy them. This term f can be generated using the ISCSO algorithm and its weight and bias must be adjusted to solve the minimization problem. Fitness function for a given input y ISCSO algorithm defines,

$$n = \sum_{j=1}^{g} v_j \sigma(w_j), \tag{38}$$

$$w_j = \sum_{i=1}^{N} w_{ji} y_i + a_j \tag{39}$$

$w_{ji}$ input unit $j$ represents the load that connects the hidden unit to $j$, $v_j$ the input unit $j$ represents the load that connects $j$ to the output unit, $a_j$ the hidden unit represents the dependence of $j$, and $\sigma(w)$ is a sigmoidal transfer function (tansig). However, when the distance between $x_i^c$ and $p_{\text{best}}^c$ is less than the radius, just deviate it from $p_{\text{best}}^c$. The value of fitness elements is distinct by,

$$U_i^c(s+1) = \omega \cdot U_i^c(t) + D_1 \cdot r_1 \cdot (p_{\text{best}}^c(t) - x_i^c(t) \tag{40}$$

To find a small solution to the problem of exercise activity, which is defined as follows:

$$Fs_{\text{a}} = Fs_{\text{max}}; \quad \text{otherwise,} \;\; Fs_{\text{a}} = Fs_{\text{min}} \tag{41}$$

The position mode in which the target is detected is called tracking mode. This process can be summarized in three steps. Update the speed of each measure according to the optimal level to transfer to the entire group of sand cats, i.e., the optimal solution currently found:

$$U_i^c(s+1) = U_i^c(t) + r \cdot d(x_{\text{best}}^c(t) - x_i^c(t)) \quad C = 1, 2, 3 \ldots m \tag{42}$$

Verify the threshold conditions, if the speed is within the maximum speed limit. Update the sand cat's position as follows:

$$x_i^c(s+1) = x_i^c(t) + U_i^c(t+1)) \tag{43}$$

This improvement not just works on the exhibition and incorporation of the calculation, yet additionally keeps a comprehension of appropriation strength. The functioning system of cluster formation and coverage hole detection using ISCSO is described in Algorithm 2.

**Algorithm 2 Cluster based coverage holes detection using ISCSO**

Input : location, total number of nodes in network
Output : clustering and coverage holes

1. Initialize the parameters such as number of sand cats, Maximum number of iterations, small radius, and deviation from the radius
2. Initialize the sand cats' positions randomly within the network coverage area
3. For each sand cat i, calculate its fitness value
4. Initialize the historical best position and fitness for each sand cat
5. While (iteration < MaxIter) Do
6 Find the index of the sand cat with the best fitness value
7. iteration = iteration + 1
8. End While
9. Identify coverage holes within each cluster by using a Hyb-DRL
10. End for
11. End

### 4.3 Hole shape detection and hole size judgment

Hole shape detection refers to the process of identifying the shape or geometry of a coverage hole in a WSN. In other words, it involves determining the boundaries or outline of the area in which there is no coverage by the sensor nodes. Hole size judgment, on the other hand, is the process of estimating the size or area of the coverage hole detected in the WSN. This information is useful for optimizing the placement of additional sensor nodes or adjusting the transmission power levels to ensure complete coverage of the network. By accurately detecting the shape and size of the coverage hole, network managers can identify the best locations to deploy new nodes or reposition existing ones to improve network coverage and reliability. Additionally, this information can be used to optimize routing protocols to ensure that data is transmitted through the most reliable and efficient path in the network. Therefore, hole shape detection and size judgment are essential components of any coverage hole detection algorithm for WSN.

The long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is well suited for processing and making predictions based on sequential data. In the context of reinforcement learning, the LSTM can be used as part of an agent's architecture to model and predict future states and rewards. In the case of determining the shape and size of holes in WSNs (wireless sensor networks), the Hyb-DRL (hybrid deep reinforcement learning) approach combines LSTM with other deep reinforcement learning techniques to address this problem. Specifically, the layer details of the Hyb-DRL model may involve the following components:

- LSTM Layers: The LSTM layers are responsible for capturing and learning sequential patterns in the input data. These layers allow the model to retain information over a certain time window, making it suitable for processing time series data.

- Deep Reinforcement Learning Layers: These layers typically include fully connected layers and other nonrecurrent layers that facilitate the training and decision-making process in reinforcement learning. They can help in learning the optimal policy for determining the shape and size of holes in WSNs based on the current state and potential rewards.
- Output Layers: The output layers of the Hyb-DRL model provide the final predictions or decisions based on the input data and learned policies. In the context of determining the shape and size of holes in WSNs, the output layers may produce the desired configurations or parameters for the network layout.

It is important to note that the specific layer details of the Hyb-DRL model may vary depending on the implementation and specific requirements of the problem. The mentioned components provide a general overview of how LSTM and reinforcement learning can be combined to tackle the challenge of determining the shape and size of holes in WSNs.

In the context of coverage hole detection in WSNs, Hyb-DRL can be used to accurately detect the shape and size of coverage holes within clusters, among clusters and along edges. The Hyb-DRL algorithm is used to identify the shape and size of coverage holes by analyzing the sensor data and making decisions based on the reward signals. The deep neural network is trained using the $Q$-learning algorithm to optimize the decision-making process and improve the accuracy of the coverage hole detection. Hyb-DRL is particularly effective in situations where there is a large amount of data and complex decision-making processes involved. By combining the principles of deep learning and reinforcement learning, Hyb-DRL can identify complex patterns and make accurate decisions, making it a useful tool for coverage hole detection in WSNs. To update the gate's value affects how much information is brought in from the previous moment. We can determine the update fitness by:

$$z_s = \sigma(W_z \cdot [G_{s-1}, Y_s] + A_z) \tag{44}$$

where $w_z$ and $a_z$ are the update gate's bias and weight matrix, respectively. "$(X) = 1/[1 + \exp(X)]$" denotes the Sigmoid function, which serves as the gate-control signal by transforming the data into values between 0 and 1. The reset door is utilized to control the amount of the concealed layer data from the past second should be failed to remember you can sort it out by:

$$r_s = \sigma(W_R \cdot [G_{s-1}, Y_s] + A_R) \tag{45}$$

where $W_R$ and $A_R$ are the reset gate's bias and weight matrix, respectively. The sigmoid will set the reset gate's output to 0 to erase previous moment's information about the hidden state.

$$\tilde{G}_s = \tan G(W_g \cdot [G_s \circ G_{s-1}, Y_s] + A_g) \tag{46}$$

where $W_R$ and $A_R$ are, respectively, the candidate output state's bias and weight matrix, and tanh calls the function that scales the data between 0 and 1. Hyb-DRL output layer contains the ideal secret layer state, not entirely set in stone by and. The following is the mathematical expression:

$$G_s = (1 - z_s) \circ G_{s-1} + z_s \circ \tilde{G}_s \tag{47}$$

The larger $z_s$ is, the higher level of reliance on $G_s$ on $\tilde{G}_s$ is, and ht1 plays a smaller role in determining the output.$(1 - z_s) \circ G_{s-1}$ The current node's information, $ht$, points to selective memory for the previous hidden state. The forward Hyb-DRL layer stores moment $t$ and the input sequence's previous moment, while the subsequent moment is stored in the backward Hyb-DRL layer. The process of propagating hidden layers in Hyb-DRL classifier is,

$$\underset{s}{\overset{\rightharpoonup}{G}} = f(\overset{\rightharpoonup}{W} Y_s + \overset{\rightharpoonup}{V} \overset{\rightharpoonup}{G}_{s-1} + \vec{A}) \tag{48}$$

$$\underset{s}{\overset{\leftharpoonup}{G}} = f(\overset{\leftharpoonup}{W} Y_s + \overset{\leftharpoonup}{V} \overset{\leftharpoonup}{G}_{s-1} + \overset{\leftharpoonup}{A}) \tag{49}$$

$$X_s = \sigma(\underset{s}{\overset{\rightharpoonup}{G}}, \underset{s}{\overset{\leftharpoonup}{G}}) \tag{50}$$

where $\underset{s}{\overset{\rightharpoonup}{G}}$ and $\underset{s}{\overset{\leftharpoonup}{G}}$ indicate the forward and in reverse estimation stowed away layer states, separately; mean, in both forward and reverse computations, the heaviness of the contribution as well as the previous state of the hidden layer; $\vec{A}$ and $\underline{A}$, respectively, denote the forward and backward calculation bias. Qu-bit state is used in ensemble learning. A bit state is also frequently controlled by two quantum logic gates, a controlled NOT gate with two Qu bits and a cycle gate with one bit. The state of the $I$th $kv$-bit neuron model in $M$th sets is defined as follows based on these gates:

$$Z_i^M = f\left\{ \frac{\pi}{2}\delta_i^M - \arg\left[ \sum_K f(\theta_{K,i}^M) f(z_K^{M-1}) - f(\lambda_i^M) \right] \right\} s \tag{51}$$

here $z_K^{M-1}$ is the controlled NOT gate's reversal parameter, Input of $K$th neuron in $(m-1)$ sets. This is the best step for cyclic gate phase parameters and threshold parameters. The output $O$ layer of the network is denoted by m observed state from the $j$th neuron of the output layer. $UNN_i$ is represented follows:

$$\mathrm{UNN}_i = \lfloor jm(Z_i^0) \rfloor^2 \tag{52}$$

The best parameters are sought when training the multilayer quantum neural network.$\theta_{K,i}^M$,$\delta_i^M$ and $\lambda_i^M$ that make the subsequent cost function smaller:

$$i(w) = \frac{1}{2} \sum_q \sum_i \{U_{d_i}(Q) - UNN_i(w, q)\}^2 \tag{53}$$

where $U_{d_i}$ is the instruction signal for the $i$th neuron in the $q$th pattern. New members of the ensemble crossover in the real-coded crossover $\omega_C$ are created by individuals with multiple parents.

$$\omega_C = \omega_h + \sum_{j=1}^{n+k} \xi_j(\omega_j - \omega_h) \tag{54}$$

The just generation gap model is the one that assumes that children replace parents in each generation alternation model used. The cost function's reciprocal defines the fitness function $F(\omega_j)$, where the number of people is denoted by $j$. We consider the following discrete-time factory as the target system to be controlled in the design of a direct quantum neural network controller:

$$X(k + d_q) = F_q[X(k), ..., X(k - n_q + 1), u(k), ..., u(k - m_q - d_q + 1)] \tag{55}$$

where $x$ is the factory output, $U$ is the factory input, factory commands, $K$ is the model number and the factory idle time is a function describing it dynamic control variable. Algorithm 3 describes the working process involved in the coverage hole shape and hole size judgment.

**Algorithm 3 Coverage holes shape and hole size judgment using Hyb-DRL**

Input : coverage holes, location of holes, total number of holes
Output : size and shape

1 Initialize the parameters
2 While Do
3 The hidden layer propagation process of reinforcement learning
4 Compute candidate output state $\tilde{g}_s$ by $\tilde{G}_s = \tan G(W_g \cdot [G_s \circ G_{s-1}, Y_s] + A_g)$
5 For $s = 1$, S do
6 A state superposition that is coherent as $|\psi>= B|0> + A|1>$.
7 As the system to control, compute discrete time with one input and one output.
$X(k + d_q) = F_q[X(k), ..., X(k - n_q + 1), u(k), ..., u(k - m_q - d_q + 1)]$
8 Compute decision metrics
9 End

## 5 Results and discussion

In this section, the performance of proposed optimal cluster-based node position estimation and coverage hole detection (named as OC-NP-CHD for results explanation purpose) approach is validated through different simulation scenarios and measures. Our proposed OC-NP-CHD approach is implement and simulated using NS-2.33 simulation tool. Besides, in order to justify the effectiveness of our proposed OC-NP-CHD approach in terms of simulation results, the reenactment results are contrasted and the benchmark inclusion opening location draws near, for example, DHC [27], PS [28], DCHD [29] and CG-DCHD [26].

### 5.1 Simulation setup

Table 2 provides a detailed description of the simulation setup and parameters used for validating the proposed OC-NP-CHD approach. The network size is set to $500\times500$ m$^2$, which is a reasonable size for a wireless sensor network. The number of sensor nodes is varied from 600 to 1000 in increments of 100 to evaluate the performance of the proposed approach under different network densities. The number of simulation rounds is varied from 1000 to 5000 to obtain statistically significant results. The number of coverage holes is varied from 10 to 50 in increments of 10 to evaluate the ability of the proposed approach to detect coverage holes of different sizes. The sensing range and communication range of each sensor node are set to 20 m to simulate the radio coverage of the wireless sensor network. The IEEE 802.15.4 MAC/PHY specification is used for the simulation, and the ad hoc on-demand distance vector (AODV) routing protocol is used for routing packets. The initial energy of each sensor node is set to 10 J, which is a reasonable value for a wireless sensor node with a battery-powered energy source. The energy cost for control packets is set to 0.3 J to simulate the energy consumption of control packets in the network. The data rate is set to 250 Kbps, which is a reasonable value for a wireless sensor network. The simulation is run for a total time of 500 s, which provides sufficient time to evaluate the performance of the proposed approach.

### 5.2 Comparative analysis

In this section, the simulation results and comparative analysis of proposed and existing coverage hole detection approaches with respect to three different simulation scenarios, such as impact of node density, impact of hole density and impact of simulation rounds.

**Table 2** Simulation setup and parameters

| Parameter | Value |
|---|---|
| Network size | $500\times500$ m$^2$ |
| Number of sensor nodes | 600, 700, 800, 900 and 1000 |
| Number of simulation rounds | 1000, 2000, 3000, 4000 and 5000 |
| Number of holes | 10, 20, 30, 40 and 50 |
| Sensing range | 20 m |
| Communication range | 20 m |
| MAC/PHY | IEEE 802.15.4 |
| Routing | AODV |
| Initial energy of each sensor node | 10 J |
| Energy cost for control packets | 0.3 J |
| Data rate | 250 Kbps |
| Total simulation time | 500 s |

### 5.2.1 Impact of node density

In this scenario, we vary the number of nodes as 600, 700, 800, 900 and 1000 with the fixed network size as $500 \times 500$ m$^2$ area. Table 3 describes the comparative analysis of proposed and existing coverage hole detection approaches with respect to impact of node density. The average energy consumption of our proposed OC-NP-CHD approach is 65.339%, 56.539%, 41.752% and 11.71% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 2 shows the average energy consumption results of proposed and existing coverage hole detection approaches with respect to impact of node density. The hole detection time of our proposed OC-NP-CHD approach is 32.174%, 24.458%, 14.761% and 2.207% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 3 shows the hole detection time results of proposed and existing coverage hole detection approaches with respect to impact of node density. The coverage of our proposed OC-NP-CHD approach is 9.474%, 7.368%, 5.263% and 45.433% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 4 shows the coverage results of proposed and existing coverage hole detection approaches with respect to impact of node density. The network lifetime of our proposed OC-NP-CHD approach is 19.175%, 16.779%, 14.382% and 11.985% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 5 shows the network lifetime results of proposed and existing coverage hole detection approaches with respect to impact of node density. The control packet overhead of our proposed OC-NP-CHD approach is 1.453%, 1.04%, 0.623% and 0.203% minimized compared to the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively.

### 5.2.2 Impact of hole density

In this scenario, we vary the number of holes as 10, 20, 30, 40 and 50 with the fixed number of nodes as 1000 and network size as $500 \times 500$ m$^2$ area. Table 4 describes the comparative analysis of proposed and existing coverage hole detection approaches with respect to impact of hole density. The average energy consumption of our proposed OC-NP-CHD approach is 65.339%, 56.539%, 41.752% and 11.71% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 6 shows the average energy consumption results of proposed and existing coverage hole detection approaches with respect to impact of hole density. The hole detection time of our proposed OC-NP-CHD approach is 32.174%, 24.458%, 14.761% and 2.207% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 7 shows the hole detection time results of proposed and existing coverage hole detection approaches with respect to impact of hole density. The coverage of our proposed OC-NP-CHD approach is 9.474%, 7.368%, 5.263% and 45.433% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 8 shows the coverage results of proposed and existing coverage hole detection approaches with respect to impact of hole density. The network lifetime of our proposed

**Table 3** Results of proposed and existing coverage hole detection approaches with respect to impact of node density

| Coverage hole detection approaches | Average energy consumption (J) | | | | | Hole detection time (sec) | | | | | Coverage (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 600 | 700 | 800 | 900 | 1000 | 600 | 700 | 800 | 900 | 1000 | 600 | 700 | 800 | 900 | 1000 |
| DHC [27] | 4.607 | 4.907 | 5.059 | 5.336 | 5.65 | 14.939 | 15.093 | 15.273 | 15.601 | 15.849 | 83 | 85 | 86 | 87 | 89 |
| PS [28] | 3.572 | 3.872 | 4.024 | 4.301 | 4.615 | 13.371 | 13.525 | 13.705 | 14.033 | 14.281 | 85 | 87 | 88 | 89 | 91 |
| DCHD [29] | 2.537 | 2.837 | 2.989 | 3.266 | 3.58 | 11.803 | 11.957 | 12.137 | 12.465 | 12.713 | 87 | 89 | 90 | 91 | 93 |
| CG-DCHD [26] | 1.502 | 1.802 | 1.954 | 2.231 | 2.545 | 10.235 | 10.389 | 10.569 | 10.897 | 11.145 | 89 | 91 | 92 | 93 | 95 |
| OC-NP-CHD | 1.267 | 1.567 | 1.719 | 1.996 | 2.31 | 10 | 10.154 | 10.334 | 10.662 | 10.91 | 92 | 94 | 95 | 96 | 98 |
| | Network lifetime (sec) | | | | | Control packet overhead (kb) | | | | | | | | | |
| DHC [27] | 197 | 194 | 174 | 150 | 128 | 0.100 | 0.595 | 2.635 | 10.644 | 15.972 | | | | | |
| PS [28] | 202 | 199 | 179 | 155 | 133 | 0.075 | 0.570 | 2.610 | 10.619 | 15.947 | | | | | |
| DCHD [29] | 207 | 204 | 184 | 160 | 138 | 0.050 | 0.545 | 2.585 | 10.594 | 15.922 | | | | | |
| CG-DCHD [26] | 212 | 209 | 189 | 165 | 143 | 0.025 | 0.520 | 2.560 | 10.569 | 15.897 | | | | | |
| OC-NP-CHD | 237 | 234 | 214 | 190 | 168 | 0.013 | 0.508 | 2.548 | 10.557 | 15.885 | | | | | |

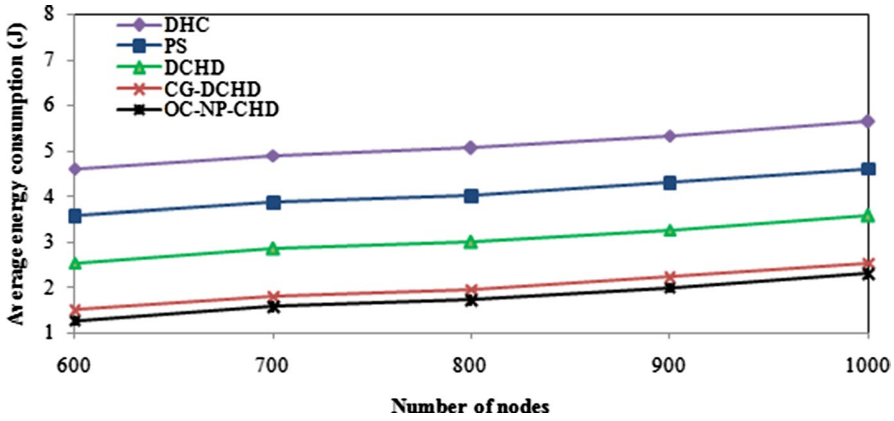


**Fig. 2** Average energy consumption with node density

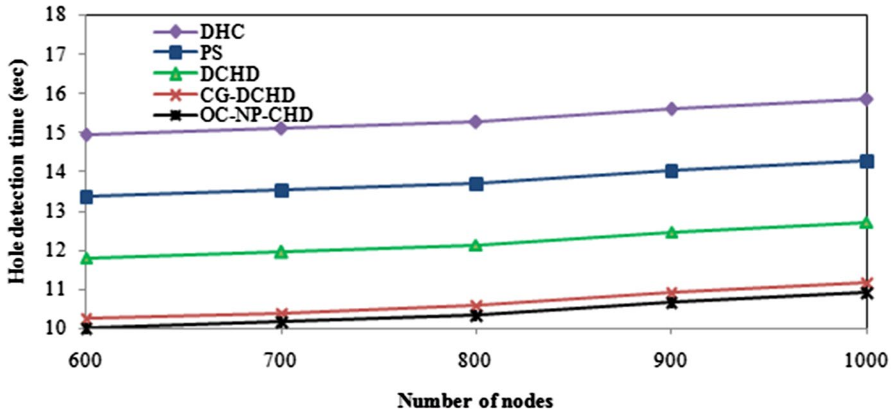


**Fig. 3** Hole detection time with node density

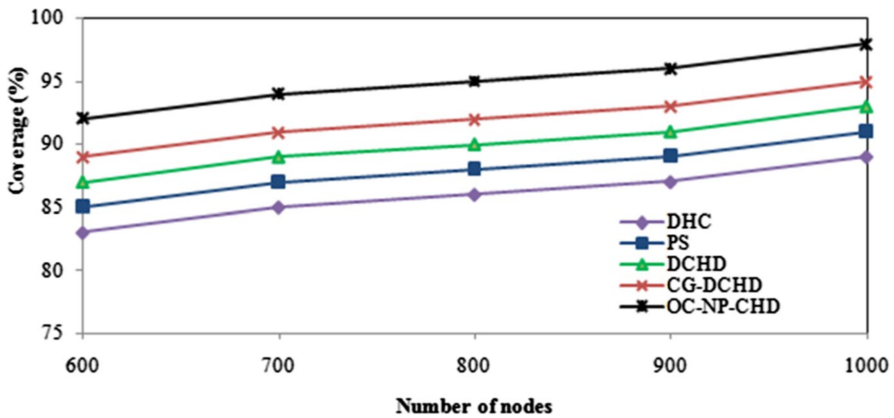


**Fig. 4** Coverage with node density

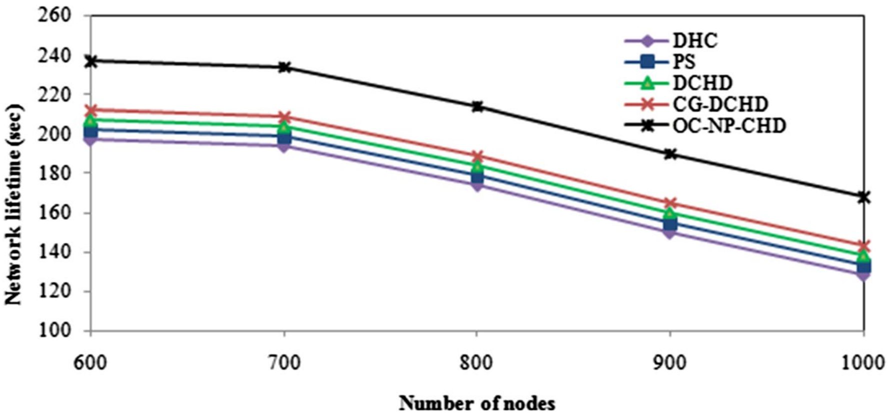


**Fig. 5** Network lifetime with node density

OC-NP-CHD approach is 19.175%, 16.779%, 14.382% and 11.985% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 9 shows the network lifetime results of proposed and existing coverage hole detection approaches with respect to impact of hole density. The control packet overhead of our proposed OC-NP-CHD approach is 1.453%, 1.04%, 0.623% and 0.203% minimized compared to the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively.

### 5.2.3 Impact of simulation rounds

In this scenario, we vary the number of holes as 10, 20, 30, 40 and 50 with the fixed number of nodes as 1000 and network size as $500 \times 500$ m$^2$ area. Table 5 describes the comparative analysis of proposed and existing coverage hole detection approaches with respect to impact of simulation rounds. The average energy consumption of our proposed OC-NP-CHD approach is 65.339%, 56.539%, 41.752% and 11.71% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 10 shows the average energy consumption results of proposed and existing coverage hole detection approaches with respect to the impact of simulation rounds. The hole detection time of our proposed OC-NP-CHD approach is 32.174%, 24.458%, 14.761% and 2.207% lower than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively.

Figure 11 shows the hole detection time results of proposed and existing coverage hole detection approaches with respect to impact of simulation rounds. The coverage of our proposed OC-NP-CHD approach is 9.474%, 7.368%, 5.263% and 45.433% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 12 shows the coverage results of proposed and existing coverage hole detection approaches with respect to impact of simulation rounds. The network lifetime of our proposed OC-NP-CHD approach is 19.175%, 16.779%, 14.382% and 11.985% higher than the existing benchmark approaches are CG-DCHD, DHC, PS and DCHD, respectively. Figure 13 shows the network lifetime

**Table 4** Results of proposed and existing coverage hole detection approaches with respect to impact of hole density

| Coverage hole detection approaches | Average energy consumption (J) | | | | | Hole detection time (sec) | | | | | Coverage (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 |
| DHC [27] | 5.932 | 6.232 | 6.384 | 6.661 | 6.975 | 16.264 | 16.418 | 16.598 | 16.926 | 17.174 | 81 | 83 | 84 | 85 | 87 |
| PS [28] | 4.897 | 5.197 | 5.349 | 5.626 | 5.94 | 14.696 | 14.850 | 15.030 | 15.358 | 15.606 | 83 | 85 | 86 | 87 | 89 |
| DCHD [29] | 3.862 | 4.162 | 4.314 | 4.591 | 4.905 | 13.128 | 13.282 | 13.462 | 13.79 | 14.038 | 85 | 87 | 88 | 89 | 91 |
| CG-DCHD [26] | 2.827 | 3.127 | 3.279 | 3.556 | 3.87 | 11.56 | 11.714 | 11.894 | 12.222 | 12.47 | 87 | 89 | 90 | 91 | 93 |
| OC-NP-CHD | 2.592 | 2.892 | 3.044 | 3.321 | 3.635 | 11.325 | 11.479 | 11.659 | 11.987 | 12.235 | 90 | 92 | 93 | 94 | 96 |
| | Network lifetime (sec) | | | | | Control packet overhead (kb) | | | | | | | | | |
| DHC [27] | 172 | 169 | 149 | 125 | 103 | 2.663 | 3.158 | 5.198 | 13.207 | 18.535 | | | | | |
| PS [28] | 177 | 174 | 154 | 130 | 108 | 2.638 | 3.133 | 5.173 | 13.182 | 18.51 | | | | | |
| DCHD [29] | 182 | 179 | 159 | 135 | 113 | 2.613 | 3.108 | 5.148 | 13.157 | 18.485 | | | | | |
| CG-DCHD [26] | 187 | 184 | 164 | 140 | 118 | 2.588 | 3.083 | 5.123 | 13.132 | 18.46 | | | | | |
| OC-NP-CHD | 212 | 209 | 189 | 165 | 143 | 2.463 | 2.958 | 4.998 | 13.007 | 18.335 | | | | | |

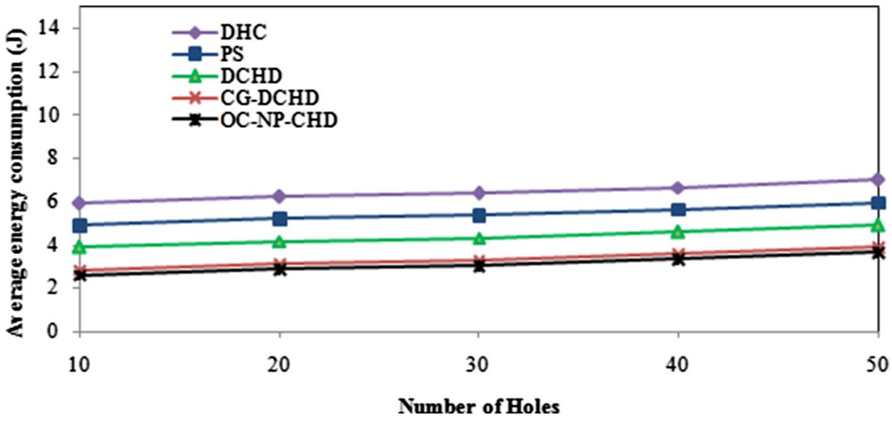


**Fig. 6** Average energy consumption with hole density

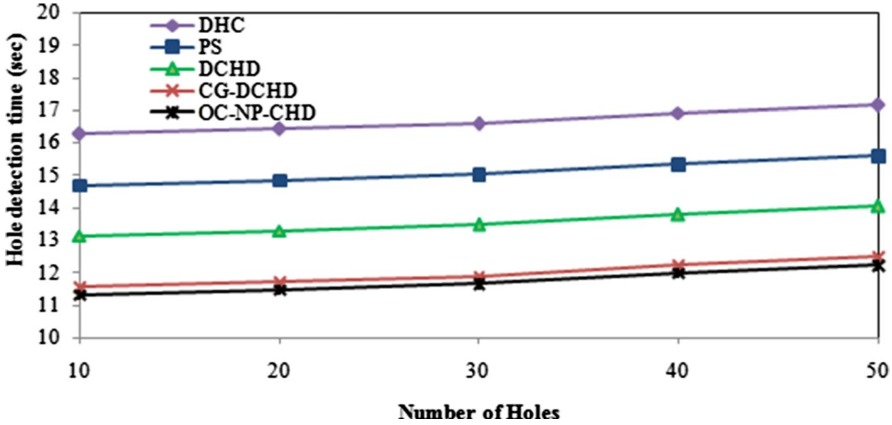


**Fig. 7** Hole detection time with hole density

results of proposed and existing coverage hole detection approaches with respect to impact of simulation rounds. The control packet overhead of our proposed OC-NP-CHD approach is 1.453%, 1.04%, 0.623% and 0.203% minimized compared to the existing benchmark approaches are CG-DCHD, DHC, PS andDCHD, respectively.

## 6 Conclusion

The computational geometry-based approaches are failed to detect coverage holes between sensors and boundary of the region. We have proposed an optimal cluster-based node position estimation and coverage hole detection (OC-NP-CHD) approach to solve the problems in the previous CG-DCHD approach. The MLO algorithm is used for computing the node positions in the WSN. Its impact is to

**Table 5** Results of proposed and existing coverage hole detection approaches with respect to impact of simulation rounds

| Coverage hole detection approaches | Average energy consumption (J) | | | | | Hole detection time (sec) | | | | | Coverage (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1000 | 2000 | 3000 | 4000 | 5000 | 1000 | 2000 | 3000 | 4000 | 5000 | 1000 | 2000 | 3000 | 4000 | 5000 |
| DHC [27] | 7.257 | 7.557 | 7.709 | 7.986 | 8.3 | 17.589 | 17.743 | 17.923 | 18.251 | 18.499 | 79 | 81 | 82 | 83 | 85 |
| PS [28] | 6.222 | 6.522 | 6.674 | 6.951 | 7.265 | 16.021 | 16.175 | 16.355 | 16.683 | 16.931 | 81 | 83 | 84 | 85 | 87 |
| DCHD [29] | 5.187 | 5.487 | 5.639 | 5.916 | 6.23 | 14.453 | 14.607 | 14.787 | 15.115 | 15.363 | 83 | 85 | 86 | 87 | 89 |
| CG-DCHD [26] | 4.152 | 4.452 | 4.604 | 4.881 | 5.195 | 12.885 | 13.039 | 13.219 | 13.547 | 13.795 | 85 | 87 | 88 | 89 | 91 |
| OC-NP-CHD | 3.917 | 4.217 | 4.369 | 4.646 | 4.96 | 12.65 | 12.804 | 12.984 | 13.312 | 13.56 | 88 | 90 | 91 | 92 | 94 |
| | Network lifetime (sec) | | | | | Control packet overhead (kb) | | | | | | | | | |
| DHC [27] | 147 | 144 | 124 | 100 | 78 | 5.226 | 5.721 | 7.761 | 15.77 | 21.098 | | | | | |
| PS [28] | 152 | 149 | 129 | 105 | 83 | 5.201 | 5.696 | 7.736 | 15.745 | 21.073 | | | | | |
| DCHD [29] | 157 | 154 | 134 | 110 | 88 | 5.176 | 5.671 | 7.711 | 15.72 | 21.048 | | | | | |
| CG-DCHD [26] | 162 | 159 | 139 | 115 | 93 | 5.151 | 5.646 | 7.686 | 15.695 | 21.023 | | | | | |
| OC-NP-CHD | 187 | 184 | 164 | 140 | 118 | 5.026 | 5.521 | 7.561 | 15.57 | 20.898 | | | | | |

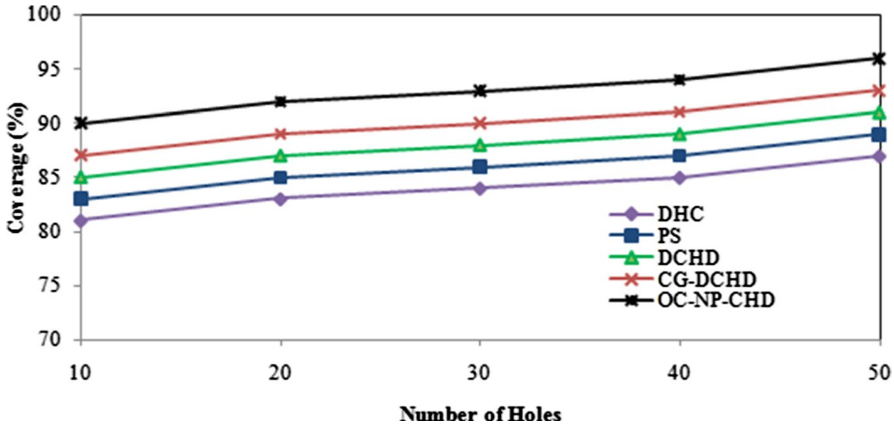


**Fig. 8** Coverage with hole density

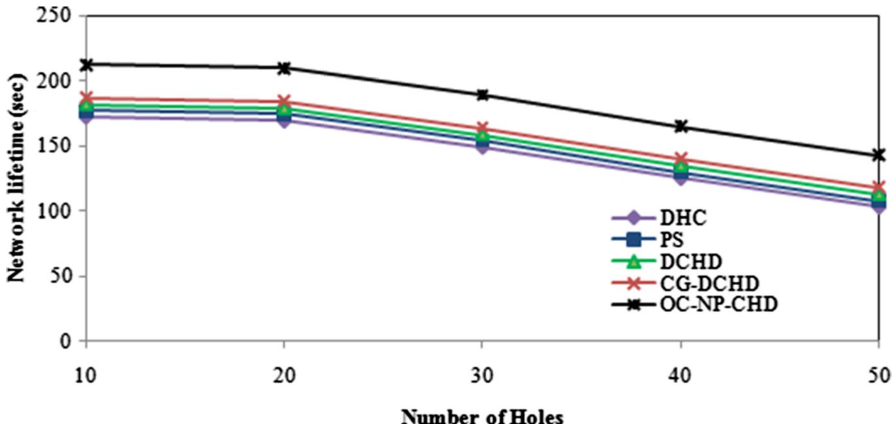


**Fig. 9** Network lifetime with hole density

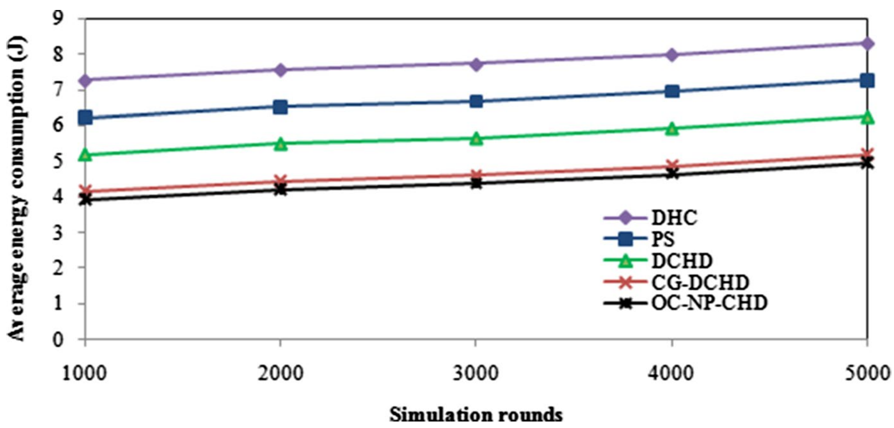


**Fig. 10** Average energy consumption with simulation rounds

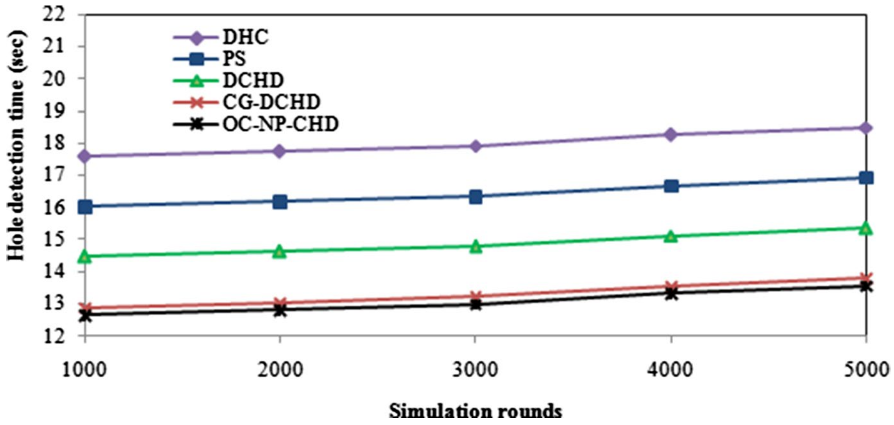


**Fig. 11** Hole detection time with simulation rounds

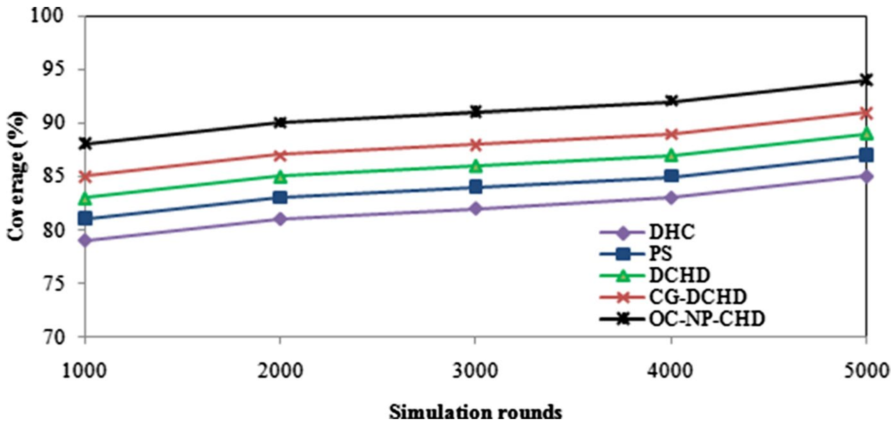


**Fig. 12** Coverage with simulation rounds

ensure that the edge nodes in the network are optimally positioned. This helps in reducing the size of coverage holes in the network and improving the overall coverage of the WSN. The ISCSO algorithm is used for optimal clustering of the sensor nodes in the network. Its impact is to balance the clusters and efficiently compute the coverage hole area in the network. This helps in detecting coverage holes in a more accurate and efficient manner. The Hyb-DRL technique is used for detecting the shape and size of coverage holes within clusters, among clusters and along edges. To validate the performance of proposed CG-DCHD approach by using different simulation scenarios and measures. From the results we observed that the average energy consumption of our CG-DCHD approach is 43.835%, 32.674% and 26.164% lower compared to the existing benchmark approaches for node density, hole density and simulation rounds, respectively. The hole detection time of our CG-DCHD

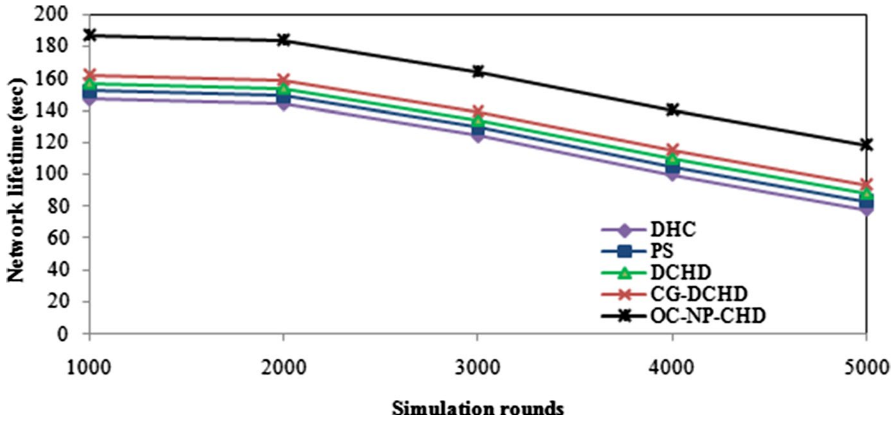


**Fig. 13** Network lifetime with simulation rounds

approach is 18.4%, 16.802% and 15.462% lower compared to the existing benchmark approaches for node density, hole density and simulation rounds, respectively. The coverage of our CG-DCHD approach is 16.885%, 14.977% and 12.219 higher compared to the existing benchmark approaches for node density, hole density and simulation rounds, respectively. The network lifetime of our CG-DCHD approach is 15.58%, 17.702% and 20.492% higher compared to the existing benchmark approaches for node density, hole density and simulation rounds, respectively. The control packet overhead of our CG-DCHD approach is 0.83%, 1.907% and 1.466% lower compared to the existing benchmark approaches for node density, hole density and simulation rounds, respectively. Energy consumption is a critical concern in WSNs. Future research could explore ways to optimize energy efficiency in the OC-NP-CHD approach. This may involve developing energy-aware algorithms, adaptive power management techniques or energy harvesting strategies to prolong the network's lifetime and reduce energy consumption.

**Author contributions** Both the Authors contributed in writing the main manuscript text and performed the simulation. Both Authors prepared figures and tables and both authors reviewed the manuscript.

## Declarations

**Competing interests** The authors declare no competing interests.

## References

1. Wang F, Hu H (2021) Coverage hole detection method of wireless sensor network based on clustering algorithm. Measurement 179:109449
2. Khedr AM, Osamy W, Salim A (2018) Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks. Comput Commun 124:61–75

3. Li W, Zhang W (2015) Coverage hole and boundary nodes detection in wireless sensor networks. J Netw Comput Appl 48:35–43
4. Gou P, Mao G, Zhang F, Jia X (2020) Reconstruction of coverage hole model and cooperative repair optimization algorithm in heterogeneous wireless sensor networks. Comput Commun 153:614–625
5. Zygowski C, Jaekel A (2020) Optimal path planning strategies for monitoring coverage holes in Wireless Sensor Networks. Ad Hoc Netw 96:101990
6. Chowdhury A, De D (2021) Energy-efficient coverage optimization in wireless sensor networks based on Voronoi–Glowworm swarm optimization-K-means algorithm. Ad Hoc Netw 122:102660
7. Deng X, Xu M, Yang LT, Lin M, Yi L, Wang M (2018) Energy balanced dispatch of mobile edge nodes for confident information coverage hole repairing in IoT. IEEE Internet Things J 6(3):4782–4790
8. Das S, Debbarma MK (2020) CHPT: an improved coverage-hole patching technique based on tree-center in wireless sensor networks. J Ambient Intell Human Comput 14:5873–5884
9. Priyadarshi R, Gupta B (2020) Coverage area enhancement in wireless sensor network. Microsyst Technol 26(5):1417–1426
10. NilsazDezfouli N, Barati H (2020) A distributed energy-efficient approach for hole repair in wireless sensor networks. Wirel Netw 26(3):1839–1855
11. Deng X, Jiang Y, Yang LT, Lin M, Yi L, Wang M (2019) Data fusion based coverage optimization in heterogeneous sensor networks: a survey. Inf Fusion 52:90–105
12. Mharsi N, Hadji M (2019) A mathematical programming approach for full coverage hole optimization in Cloud Radio Access Networks. Comput Netw 150:117–126
13. Koriem SM, Bayoumi MA (2020) Detecting and measuring holes in wireless sensor network. J King Saud Univ Comput Inf Sci 32(8):909–916
14. Yi L, Deng X, Zou Z, Ding D, Yang LT (2018) Confident information coverage hole detection in sensor networks for uranium tailing monitoring. J Parallel Distri Comput 118:57–66
15. Boukerche A, Sun P (2018) Connectivity and coverage based protocols for wireless sensor networks. Ad Hoc Netw 80:54–69
16. Fang W, Song X, Wu X, Sun J, Hu M (2018) Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks. Inf Fusion 41:25–36
17. Wang Y, Wu S, Chen Z, Gao X, Chen G (2017) Coverage problem with uncertain properties in wireless sensor networks: a survey. Comput Netw 123:200–232
18. Phoemphon S, So-In C, Leelathakul N (2020) A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks. Expert Syst Appl 143:113044
19. Jia Y, He P, Huo L (2020) Wireless sensor network monitoring algorithm for partial discharge in smart grid. Electr Power Syst Res 189:106592
20. Xu H (2020) Semi-supervised manifold learning based on polynomial mapping for localization in wireless sensor networks. Signal Process 172:107570
21. Khalifa B, Al Aghbari Z, Khedr AM (2021) A distributed self-healing coverage hole detection and repair scheme for mobile wireless sensor networks. Sustain Comput: Inform Syst 30:100428
22. Vishnupriya G, Ramachandran R (2021) Rabin-Karp algorithm based malevolent node detection and energy-efficient data gathering approach in wireless sensor network. Microprocess Microsyst 82:103829
23. Roy S, Mazumdar N, Pamula R (2021) An energy optimized and QoS concerned data gathering protocol for wireless sensor network using variable dimensional PSO. Ad Hoc Netw 123:102669
24. Das S, KantiDebBarma M (2018) Computational geometry based coverage hole-detection and hole-area estimation in wireless sensor network. J High Speed Netw 24(4):281–296
25. Christopher VB, Jasper J (2021) Jellyfish dynamic routing protocol with mobile sink for location privacy and congestion avoidance in wireless sensor networks. J Syst Architect 112:101840
26. Ma HC, Sahoo PK, Chen YW (2011) Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks. J Netw Comput Appl 34(5):1743–1756
27. Watfa MK, Commuri S (2006) Power conservation approaches to the border coverage problem in wireless sensor networks. In: ICWN, 2006 June. pp 143–152
28. Corke P, Peterson R, Rus D (2007) Finding holes in sensor networks. In: Proceedings of the workshop on omniscient space: robot control architecture geared toward adapting to dynamic environments at ICRA April 2007
29. Kumar Sahoo P, Chiang MJ, Wu SL (2016) An efficient distributed coverage hole detection protocol for wireless sensor networks. Sensors 16(3):386

## Authors and Affiliations

**Rajib Chowdhuri[1] · Mrinal Kanti Deb Barma[1]**

✉ Rajib Chowdhuri
rjbchwdhri@gmail.com

Mrinal Kanti Deb Barma
mkdb06@gmail.com

[1] NIT Agartala, Agartala, Tripura, India