

## *Retraction*

# **Retracted: Wireless Sensor Network Architecture Based on Mobile Edge Computing**

## **Security and Communication Networks**

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

## **References**

- [1] M. Al-Razgan and T. Alfakih, "Wireless Sensor Network Architecture Based on Mobile Edge Computing," *Security and Communication Networks*, vol. 2022, Article ID 9073220, 16 pages, 2022.

## Research Article

# Wireless Sensor Network Architecture Based on Mobile Edge Computing

Muna Al-Razgan <sup>1</sup> and Taha Alfakih <sup>2</sup>

<sup>1</sup>Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11345, Saudi Arabia

<sup>2</sup>Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11345, Saudi Arabia

Correspondence should be addressed to Taha Alfakih; [talfakih@ksu.edu.sa](mailto:talfakih@ksu.edu.sa)

Received 3 February 2022; Accepted 8 March 2022; Published 1 April 2022

Academic Editor: Muhammad Arif

Copyright © 2022 Muna Al-Razgan and Taha Alfakih. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSN) are infrastructure-less and fully distributed systems of self-configurable and self-organized nodes that wish to share the IoT devices' data over the air. WSN consists of heterogeneous/homogeneous sensors dispersed in a particular area to monitor phenomena, such as volcanoes, pressure, sound, temperature, and pollution, then transfer their data to a central location for analysis and decision-making. The deployment of WSNs is often used in difficult places, such as disaster areas, battlefields, conflict zones, and volcanic zones, which require efficient data collection. However, WSNs face many challenges, such as latency, multitasks, and data quality. This paper proposes an efficient solution for multiple data collection tasks exploiting mobile edge computing (MEC) technique-enabled WSN in war field and disaster areas. Firstly, we integrate WSN and MEC to produce a novel data collection framework (WSN). Secondly, we modelled the data collection process, including multiple sensors and tasks. Thirdly, determine the best way to deploy sensors in the targeted area (e.g., military areas, isolated areas, and areas infested with terrorists) to cover the whole destination area, based on set values for data quality. Finally, find an appropriate technique for routing by choosing the routing protocol to improve their capabilities performance and decrease the energy consumption to increase the life of the WSN. Furthermore, the security of WSN is one of the key parameters, especially in the routing protocol, since it is critical to ensure the security of transmitted data and how it is transmitted securely to mitigate the attacks. The framework which works as a simulation tool shows a good result and has been proven the efficient interaction between the MEC and WSN. We compare the results with traditional methods, which proved to outperform them.

## 1. Introduction

Wireless sensors networks can be viewed as fully distributed, infrastructure-less, and self-configurable systems and self-organizing nodes for sharing data over the air. Nodes could be PCs, mobile phones, printers, and so on. WSN consists of homogeneous/heterogeneous sensors dispersed in a particular area to monitor phenomena, such as pressure, pollution, temperature, and sound, then transfer their data to a central location for analysis and decision-making. There are several challenges with WSN, such as their characteristics (variable capacity, dynamic topology, limited computing, infrastructure-free, and battery life). WSN also deploys in

places such as volcanic zones and battlefields where it is not easy to reach and without access to power, which presents yet another challenge.

We present most concepts of WSN and how they work and discuss how deployment, routing protocols, clustering, fault tolerance, and a designed WSN simulation tool will be used to understand the WSN's capabilities and limitations and help students learn how the WSN application work. Additionally, increasing the WSN capabilities by integrating them with MEC. Many challenges face the WSN to the site management of sensors; most protocols suppose the sensor contains GPS to find out more about their locations and can achieve the accuracy of the site, and it is not cost-effective,

which is equipped with every sensor with a GPS receiver. In order to achieve the deployment strategy, we will reduce the sensors' numbers required due to the issues associated with getting the sensors to change or recharge their batteries. There are two problems that will be highlighted by this challenge: first, coverage hole, which means the sensor radius will not cover some areas. The second is a connectivity hole, which means some nodes cannot connect and reach the sink. Characteristic of network: the connection between all nodes and the sink could be direct or indirect, depending on the network's structure. Scalability: any WSN consists of hundreds of nodes, so routing protocols should work with this quantity of nodes [1]. Requirements of sensing application: in this situation, the sensor will be classified depending on the application. The information collected by the sensors in WSNs may contain duplicate data (redundancy).

Wireless Sensor Networks (WSNs) are one of the IoT types since they use the Internet to communicate with several sensors. Smart gateway applications, for instance, are capable of providing services to the user on the Internet via sensors, gateways, and mobile interfaces. The security of WSN is one of the key parameters, especially in the routing protocol, since it is critical to ensure the security of transmitted data and how it is transmitted securely to mitigate the attacks. In [2], the authors proposed a WSN-based industrial monitoring system with high reliability while low power consumption. The medical application market has seen explosive growth over the past few years with the population increase. As a result of the widespread application of emerging technologies in the medical field, high-quality healthcare has been improved.

The main contribution of this work is a proposed WSN and MEC (WSNEC) framework. We aim to find an appropriate method for deploying sensors in the interested area (e.g., areas infested with terrorists and isolated areas militarised) by using an efficient algorithm. The main objectives are to enhance WSN performance mobility and reduce energy consumption. We integrated the MEC with WSN to find a suitable model for routing by selecting the suitable routing protocol communicating with relevant edge nodes to reduce the response time and provide more processing capabilities. Cloud computing is extended to the edge of the network by edge computing nodes [3].

This study is organized as follows. Section 2 introduces the related work. Section 3 presents the system model architecture. Some light is shed on the proposed framework model in Section 4. Afterward, Section 5 discusses the result analysis. Finally, we present the conclusion and future work.

## 2. Related Work

The growing interest in WSN is due to the MEMS (microelectromechanical system) technology. A sensor node consists of a data processing unit, a sensing unit, and a transfer unit. Sensors today are becoming more energy-efficient and low-cost. Sensor node networks are created due to the large size of the node, and sensors can communicate and correspond with each other [1].

There are many applications for WSN, like military applications; the sensors are widely used in military applications because sensor nodes are low-cost. Besides that, the following features are suitable for military application: robustness, self-regulation, and making error sensor networks correction [4]. Army monitoring, equipment, and ammunition, on the battlefield, cannot monitor the situation of the leaders of their armies, ammunition, and equipment from the reported data, which are collected continuously by sensor nodes and sent to the leaders [3]. Surveillance of enemy, the sensor node deploying heterogeneous groups of sensors capable of monitoring and reporting on many dynamic properties of the terrain, is crucial in a timely way. The report's information from sensor network fields is periodic and various onboard [3]. *Sensors Applications in Environment*. Many wireless sensor applications, such as flood detection and forest fire detection. *Detection of Flood*. Special sensors are equipped with sensors that alert rainfall, water levels, and weather-sensing capabilities, according to the dissemination of data over wireless connections for the sight of the light-sensitive site to the sink [4]. *Detection of Forest Fire*. Sensor network is deployed heavily as many sensor nodes scattered in a forest area and cooperate via the wireless connection. Such sensors are usually powered by solar energy produced from cells that can be used for a long time. If any node detects a fire, it immediately sends information to the sink. Home automation systems can utilize tiny sensor nodes to observe the operation of several appliances such as the refrigerator, AC, and water heater. These sensors can communicate with each other wirelessly. The system usually provides remote access and external connection to enable the control of these sensors via satellite networks or the Internet [3]. *Open Secure Office*. Sensors could be connected to most expensive office equipment such as laptops, printers, and PDAs so that when someone tries to steal them, the sensor sends a warning to the local sink that will immediately send a command to the camera to get a photo of that office [5].

David J. Stein and Esq designed the WSN simulator [6]. The network must be deployed during simulation, and simulation must be run. Before deploying the network, set the network parameters, including the number of sensors, sensor period, sensor radius, transmit cost, and cost, receive cost, transmitter period, and directed or random routing [7].

David J. Stein and Esq [6] designed the simulator and its deployment method, which limited the density of sensors; only 400 can be used. The area is fixed that cannot be changed. The sensors will start focusing on a specific area when the number of sensors decreases. It does not matter whether the sensor radius increases or decreases; the number of sensors remains the same, which leads to data redundancy and energy waste.

The simulator proposed in [6] was written by MS C#.NET 2003. Several ideas were taken from this simulator and its operation method. To identify the location of each sensor, we added an ID number for each sensor (Active or Dead). We have developed a numbering algorithm in the deployment method and found several appropriate sensors based on the available sensors and the interesting area. We

have added the flooding technique and gossiping technique for the routing approach.

The concept of edge computing is a new paradigm for IoT data processing that transfers complex computations from the cloud to the edge of networks and devices [8]. Cloud computing cannot replace edge computing as it focuses on balancing communications latency with accurate computations across the wider Internet of things hierarchy [9]. The edge nodes can support local sensing with data preprocessing capabilities and rapid reaction times in the direction of accessing the cloud. Data aggregated, filtered, and preprocessed is sent selectively to the cloud depending on application needs [10]. By distributing access to a wide variety of applications under the macrocontrol of the cloud, edge nodes can deploy a broad range of applications under the cloud's control. In addition to reducing response time in IoT communications and upload bandwidth to the cloud, edge computing enables real-time interaction at the edge and cloud-based predictive analysis [11].

MEC computing approaches lack improved interoperability between future 5G wireless sensor networks (WSN) and current wireless sensor networks (MEC). With the 5G-MEC paradigm, the author proposes an interoperable architecture that combines a wireless sensor network with a functional but outdated wireless sensor network [12]. As a result, MEC and end-users will take advantage of low latency and high transmission speeds. On the other hand, alternative transmission methods will continue to be used where 5G will not be available (white areas) or where the conditions are more difficult. The authors propose a new paradigm to improve data processing efficiency at the sensor, network, and end-user device levels by making better local resources [12].

This system collects bioinformation, as well as other relevant details such as patient image and environmental temperature. Lastly, WSN connectivity and RFID technology were used to develop a fall monitoring and accurate positioning system for the elderly [13].

### 3. Framework Architecture

The system architecture consists of the WSN architecture and the MEC architecture. We will detail each part respectively. The WSN and MEC are considered two separate networks, and each network is connected to the other through the gateways. The sensors gateway is linked to the MEC gateway, which guarantees the connection and communication between the two networks.

**3.1. Wireless Sensor Networks Architecture.** The nodes in WSN can be separated into two types, sensor node and base station node (sink), as shown in Figure 1. Firstly, *sensor nodes* are a set of nodes that move freely to monitor the physical environment, produce a packet of data, and send it to the BS (sink). Secondly, the *sink* node gathers the collected data from sensor nodes, and the users use this data packet to analyze the objectives and decision-making.

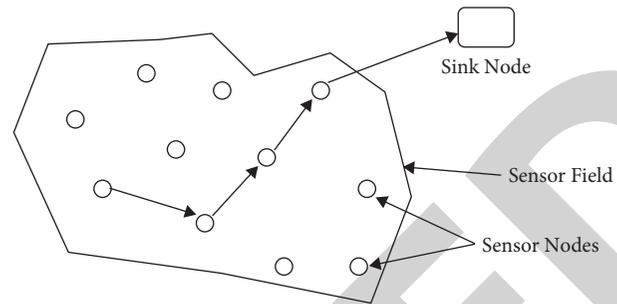


FIGURE 1: Wireless sensor node structure.

The components of the sensor [4] are divided into two parts, as shown in Figure 2, the main components and optional components. The main components include the processing unit, which is responsible for processing the received data and sending it to the sensor's transceiver unit. Transceiver unit is responsible for receiving and sending data. The sensing unit is responsible for the motion-sensing of the sensor's environment. The power unit, an essential component in the node, is considered one of the main aims for the researcher to increase the lifetime of the sensor. The location finding system is an optional component used to determine the sensor's location. Mobilizer is responsible for controlling the movement of the sensor from one location to another. And the power generator is used to recharge the sensor energy to increase the lifetime of the sensor, such as solar.

**3.2. MEC Architecture.** A distributed MCC system embeds computing resources within the RAN, close to the user's device. Figure 2 illustrates the proposed framework to integrate WSNs and MEC. The MEC hosts are computing equipment installed at or near base stations (BS) in this framework. The mobile edge host provides local virtual machines (VMs) with lower latency than the remote cloud centers. The MEC consists of a gateway that connects the main gateway of the WSN, which receives the data collected from the sensors field and directs it to the BS, which processes stores, and sends it to the monitoring unit. Each sensor node consists of the sensing unit responsible for sensing the data in the target field, the processing unit filtering the sensed data, the transmission unit to transfer data to the sensor gateway, and providing the power to the sensor node by the power unit.

Additionally, the sensor node location is determined by the position finding unit. In MEC hosts, the virtualization infrastructure within mobile edge hosts supports mobile edge applications running as VMs. The computational jobs are executed, radio network information is provided, bandwidth is managed, and UE positions are reported.

**3.3. Clustering Approach.** Clustering is a technique used to organize the nodes into groups to provide a suitable framework for developing essential features (i.e., mobility, bandwidth, allocation, routing, and topology management). Cluster heads (CH) are determined by selecting a single node

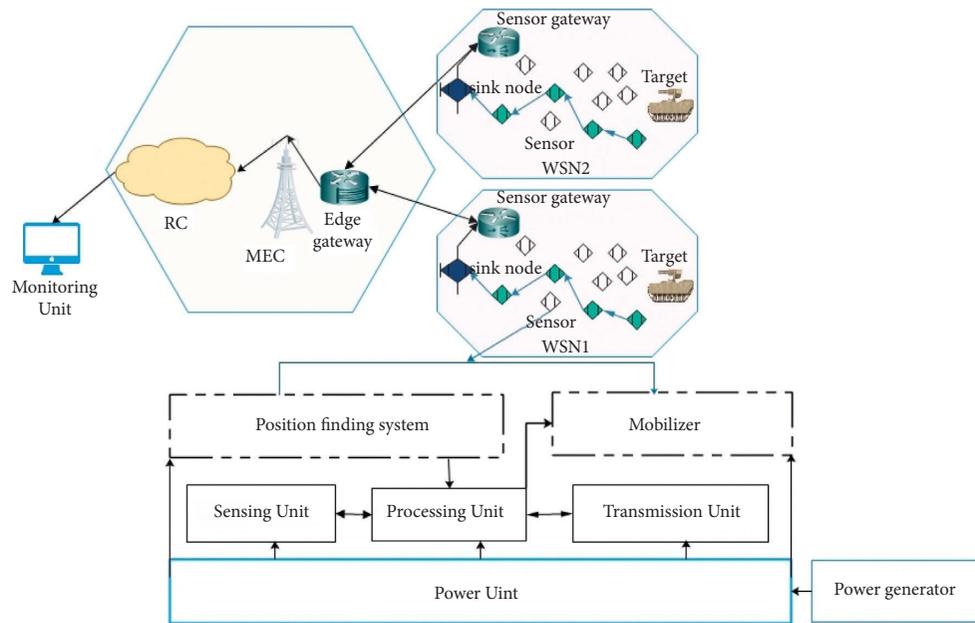


FIGURE 2: The architecture WSNEC proposed framework.

[14]. A WSN can be arranged into groups using clusters, with each group directly linked to the CH. Cluster sensors can connect directly with their corresponding CH without intermediary sensor nodes. The CHs can transmit gathered data back to the MEC through the sensor gateway, as shown in Figure 3. As a result of the clustering of sensor nodes, a consolidated set of sensed data is generated, energy consumption is reduced, and traffic and contention between clusters are reduced [15].

In clustering, the goal is to build interconnected clusters covering the entire node population. Distributed clustering algorithms can be divided into two families. According to the lowest ID algorithm, the CH is selected as the neighborhood node with the lowest ID. The second algorithm (degrees of connectivity) has the highest level of connectivity. In this status, a neighborhoods' highest degree of connectivity (degree) becomes a CH. Clustering must consider stability. The cluster must update its CH each time a node with the lowest ID moves into other clusters with the most mobility. Furthermore, when a node with the highest connectivity moves into other clusters, the cluster must reelect its CH [14].

**3.3.1. Mobility-Based Clustering Protocol (MBC).** MBC is a recent protocol on clustering for WSN. It is based on the algorithm of mobility hierarchical clustering that may produce variable-size clusters based on the mobility characteristic of the nodes [15]. As previously stated, MBC proposes that sensor nodes elect themselves as CHs depending on their mobility and remaining power. Non-cluster-head nodes aim to establish links with cluster heads as soon as possible based on estimated connection times during clustering. According to a time division multiple address system (TDMA), a timeslot is allocated for each node that is not a cluster head. Sensor nodes transmit their

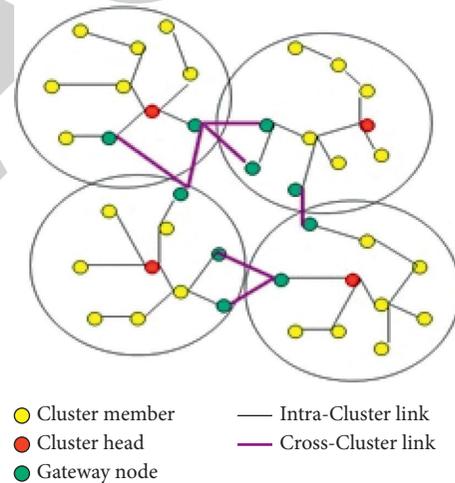


FIGURE 3: Structure of clustering.

sensed information in their timeslot to join a new cluster phase in the steady state if their connection to the previous cluster has been lost with CH; this avoids more packet loss. In addition, this protocol is designed to reduce energy usage, average control overhead, and packet loss and better adapt to highly mobile environments [16]. MBC protocol is divided into the state phase and the setup phase. To increase the successful packet delivery rate in the setup phase, a non-cluster-head node seeks a stable link to a CH.

In the setup phase, the framework attempts to increase the success rate of packet delivery by searching for a stable link to a CH. The sensor node broadcasts a joint request message in the steady-state phase to reduce packet loss when the connection time between a cluster-head and a non-cluster-head ends. In [15], the MBC steps to maintain and construct the several clusters, which are shown in Figures 4 and 5.

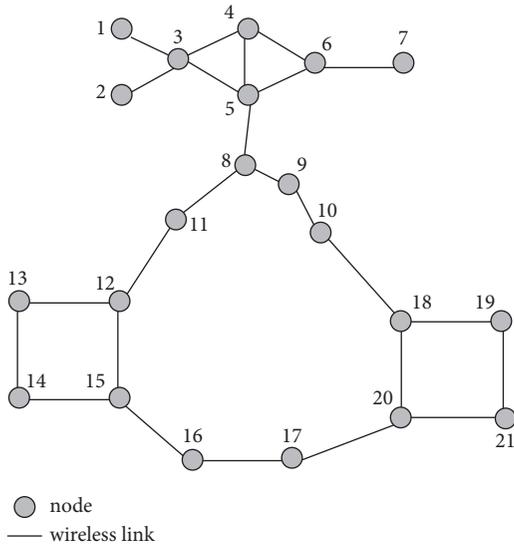


FIGURE 4: Basic concept of MBC.

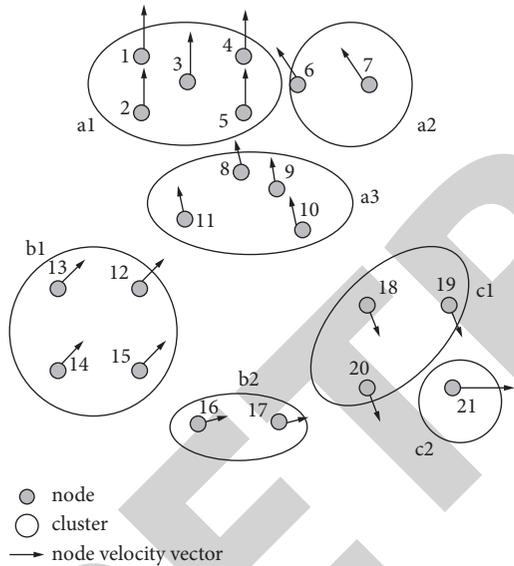


FIGURE 5: The system topology.

#### Step 1. (dissemination of mobility information).

Nodes periodically communicate their velocity information  $v(m, t)$  to their neighbors, where  $v(m, t)$  is the velocity vector at time  $t$  of node  $m$ .

**Step 2. (mobility metrics calculation).** Based on the moving information of a neighboring node  $m$  at time  $t$ , each node computes its relative mobility between nodes  $n$  and  $m$ .

$$v(m, n, t) = v(m, t) - v(n, t). \quad (1)$$

In this case, whenever two adjacent nodes are near one another, the relative mobility will be updated periodically.

#### Step 3. (construction of initial cluster).

- (1) Each node's mobility varies concerning a mobility threshold  $TH_{mob}$  compared to that threshold.

- (2) In these nodes, TCH (temporary CH) is selected based on their relative mobility  $<TH_{mob}$ .
- (3) In different networks,  $TH_{mob}$  is a design parameter that controls the stability of the cluster generated.

#### Step 4. (merging of the cluster).

- (1) If  $TCH_1$  is to be included in another  $TCH_2$  to Step3.
- (2) With its current members, child  $TCH_1$  joins parent  $TCH_2$ .
- (3) After merging the  $TCH$ , there will be new  $TCH$  for the clusters.
- (4) All nodes in the new cluster will receive routing information using the new CH.

#### Step 5. (cluster reconstruction/maintenance).

- (1) The relative mobility between a node in one cluster and the cluster head  $CH < TH_{mob}$  will not change if the node moves into a different cluster; otherwise, Step 5 will be repeated.

## 4. System Model

Sensors are assumed to have a unique ID, a range for communication, a radius for sensing, and a coordinate for the region of each sensor. Additionally, each sensor also belongs to a single area.

**4.1. WSN Parameters.** A framework model begins with setting the parameters for the deployment of the sensors and determining their distance from each other. A specific equation will determine how many sensors will be deployed in this area, so the nodes' number must be appropriate and cover the entire area. The following parameters will be considered in the proposed framework: the sensors number in the network, the network size, and the number of sensors available. If the number of nodes is high value, this increases the network's density and the number of connections. Sensor range represents the range or the radius of the sensor, which a sensor can detect events and movement Figure 6. Detection delays the delay period between detection events; if the value is set low, the detection process will be almost continuous, affecting the energy, which will lead to more energy consumption. If the value is set as large, the detection process will be at wide intervals, leading to the loss of many events. So, the appropriate value must be set to avoid this problem. Detection energy is the consumed energy during the detection process, and the communication range is the range in which two nodes in the network are able to communicate with each other to exchange data. Transmission time is required to send data packets from one node to another. Setting a high value allows the user to monitor the process of the exchange of packets on the network. Transmission energy is the amount of energy consumed in

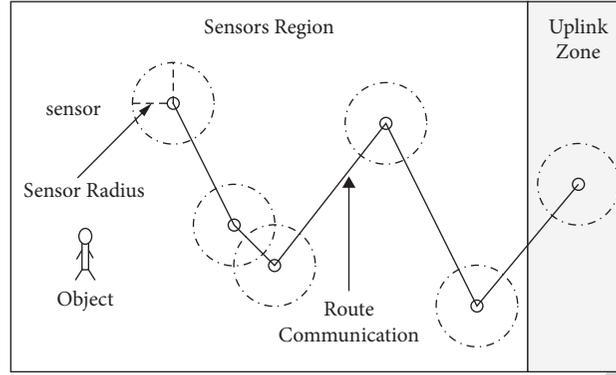


FIGURE 6: WSNEC framework model.

sending a data packet. The amount of consumed energy in the received data packet is receiving energy.

**4.2. WSN Attribute.** The parameter as mentioned earlier will be considered when selecting the sensors; moreover, there are several attributes that we should take into account, which appear in the below tables (Table 1 2).

The proposed frameworks offer flexibility in selecting parameters for sensors, including sensor radius, communications range, and sensor period.

**4.3. WSN Deployment Strategies.** Deployment of sensors is a critical issue because efficient sensors could reduce costs and enhance the wireless sensor network performance.

WSN deployment strategies have to meet different requirements, such as maintaining network connectivity, maximizing remote sensing coverage, and minimizing the use of sensor nodes. Moreover, it must be deployed in an excellent way to cover the entire field to be monitored and maintain connectivity between every sensor and the rest [17].

Deployment of sensors in the target domain directly determines the network topology, which will affect the coverage and efficiency in the WSNs to avoid two main problems of monitoring coverage (*coverage hole* and *connectivity hole*). A good deployment will increase the efficiency of resource allocation in the network and enable better performance in the collection of information and communication. At the same time, preserving topology turns out to be a difficult task because of the failure of a possible due to the depletion of energy or the destruction of [18]. Every deployment approach focuses on the implementation and maintenance of coverage and connectivity. In this section, we present some of the deployment approaches.

**4.3.1. Random Deployment.** Placement of the sensor nodes at random is the most efficient method. Random deployment is often desirable to achieve a fairly satisfactory coverage when the destination area is not accessible or prior knowledge is not available for that destination. The random deployment process for wireless network sensor nodes is feasible and practical in the military applications that throw

them from airplanes [18]. In our framework, we use random deployment; we assume deployment in the military space.

**Challenge.** There is no assurance of the coverage for the whole area we require to monitor in the random deployment. There is a possibility of uncovering parts of the area or the whole area, leading to a coverage hole problem [1]. On the other hand, some nodes are discounted or sink. This situation is called a connectivity hole [15]. The density of node can be given by [19]

$$\gamma = \frac{N}{A}, \quad (2)$$

where  $\gamma$  represents the sensor density in area  $A$  if the  $N$  indicates the number of nodes. Therefore, the deployment of highly dense sensors helps guarantee that the network connectivity and the coverage will be satisfied [15].

**4.3.2. Coverage Strategies.** There are three kinds of sensor coverage: coverage of the area, point coverage, and barrier coverage [17]. Firstly, the range of area is where every position of the field of remote sensing must be covered. Secondly, the point coverage is limited to a particular set of target points in the region of interest. Thirdly, barrier coverage minimizes the likelihood that the body passes through the checkpoint or barrier without being detected.

In our proposed framework, we want to monitor the covered interested area with a suitable number of sensors Area (Width \* Length), we calculate the sensors' number as given:

$$\text{Network Size} = \left( \frac{A}{R^2} \right), \quad (3)$$

Where  $A = \text{Width} * \text{Length}$ . (Area),

where  $R$  is the sensor radius.

The approach of deployment is randomly depending on random algorithm 1, where  $X$  and  $Y$  are the coordinates of nodes (randomly placing), and  $\text{ListSensor}[i]$  is an array of sensors (network nodes).

The sensors are deployed based on the monitoring area, and we set approximately the length and width of the

TABLE 1: Attributes with data for some nodes.

NodeID	Position	PowerLevel	Status
0	(102, 342)	1000	Active
1	(114, 356)	1000	Active
2	(205, 300)	1000	Active
3	(208, 405)	1000	Active
4	(210, 476)	1000	Active
5	(237, 360)	1000	Active
6	(239, 621)	1000	Active
7	(302, 381)	1000	Active
8	(356, 616)	1000	Active
9	(379, 402)	1000	Active
10	(399, 402)	1000	Active

TABLE 2: List of the node attributes.

Attributes	Description
NodeID	Identification of a node by its unique identifier
Position	Position of nodes in the deployment area as a function of the coordinate system X-Y
Power level	The current node's power availability
Status	The active and dead states of the nodes (a node becomes dead when the power goes out and cannot perform any sensor, sensing, communication, or computation).

destination area. The number of sensors is calculated based on the network size, considering the radius of each sensor, as equation (3). The transmission radius is considered double the sensor radius to ensure complete coverage of the whole area to be monitored, according to the analysis and results that we get from the simulation.

There is a relationship between the transmission radius for the sensor and the number of neighbors associated with this sensor; whenever the transmission radius value is significant, the number of neighbors associated with it is more, and reduce the possibility of data loss, but on the other hand that leads to increase data redundancy.

We encountered two major problems in our sensor's random deployment: connectivity and coverage problems, as illustrated in Figure 7 8. There are a few coverage holes, and another issue is the whole connectivity problem, which is almost nonexistent and much smaller than the coverage hole problem. We will reduce these problems by determining the number of the appropriate sensors based on the monitoring region and radius of the sensor. Due to the sensor's radius, coverage holes will be reduced.

Additionally, the transmission radius must be increased to reduce connectivity holes. The proposed framework is designed to automatically set the number of the sensor, allowing for a wide range of sensor configurations. Based on the entered area and the radius of each sensor, the framework automatically computes the number of sensors. There is the manual option in the framework to set any number of sensors for deployment as the other method.

**4.4. Routing Techniques.** Routes in WSNs are different from those in other traditional networks, such as computer networks. Moreover, dynamic topologies of WSN can be in various stages. This model assumes that the nodes are in a fixed location and that the deployment was random. The

sensors field sends data packets to the base station (uplink zone) using routes. There are two methods: gossiping and flooding. We also used random techniques, as described in [6]. Using the random method [6], each node is connected downstream by choosing a random path from the sensing zone to the base station until all the data packets are received. The proposed framework uses the flooding technique when the data is collected in the WSN fields and the gossiping technique when the collected data is sent to the MEC center.

**4.4.1. Gossiping Technique.** In the gossiping technique [1], nodes communicate with randomly chosen neighbors to reduce the computational load. Algorithm 2 shows that in a beginning route, messages are deployed from a sink or a node to the nearest neighbor node [20]. When the neighbor node receives this message, the node confirms the intended receiver. The routing will stop there, and if they are not the intended receiver, it chooses another neighbor node to receive this message; thus, the process is repeated recursively. All visited nodes are permanently listed using this process, preventing the message's route from becoming infinite. Figure 9 shows a red route; this route of the data is passed down from one node to another until it reaches the uplink zone. Specific parameters determine gossiping, such as residual energy, initial energy, and routing delay. The residual energy emphasizes the nodes' residual energy in making routing decisions. The initial energy emphasizes the initial energy of the nodes in routing decisions. The routing delay means the time between communication cost and packet-routing decision updates.

**4.4.2. Flooding Technique.** Flooding [1] deploys the data packets according to a traditional method, as illustrated in Algorithm 3. Flooding occurs when a packet control is transmitted from the node or the sink to each neighboring

```

Random_deployment( )
ListSensor[i];
While (Count (ListSensor[i] < NetworkSize)
While (Count (ListSensor[i] < NetworkSize)
X Random(0,Width);
Y Random(0,Height);
Node(X,Y)
Initial I = 0
While (i < Count (ListSensor[i])
If (Sensor(X,i) > NewSensor(X,i))
Insert (NewSensor(X,i), ListSensor[i]);
I+1;
End While
If (i = ListSensor[i])
Insert (NewSensor(X,i), ListSensor[i]);
End While
End

```

ALGORITHM 1: Algorithm of Random Node Deployment

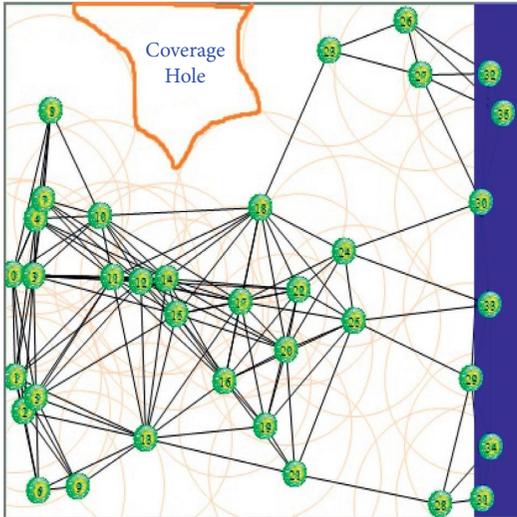


FIGURE 7: Coverage hole problem.

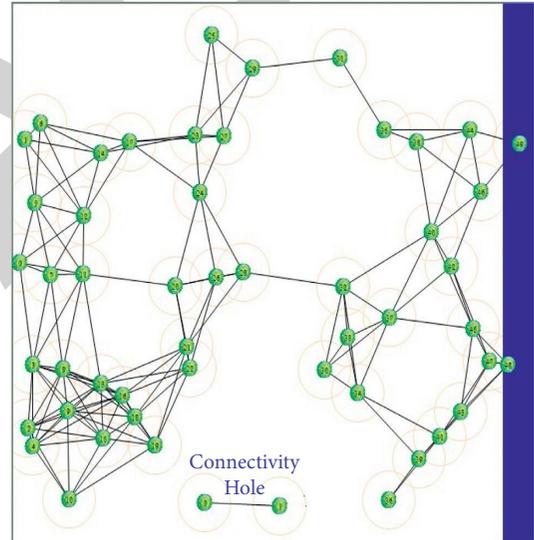


FIGURE 8: Connectivity hole problem.

node. This packet is sent by any node that receives it to all its neighbors. As a result, the message is sent to all neighbors nodes that receive it, causing the data to be repeated at each node, leading the nodes to collide. The intended recipient is the only party who knows the message's information, including the "successful" delivery status. Each node sends the same message to another, repeating the process endlessly. By maintaining all visited nodes list that received the packets, the system will prevent a message's route from forever recirculating.

**4.5. Power Consumption.** WSN's energy management has a considerable impact on their lifetime. Sensor power depletion occurs during the primary sensor's activities (sensing, communication process, and computation). The sensors consume the most energy during the communication and sensing process. This section computes the energy consumption of communication operation (data transmission and reception) in [21]. Energy can be consumed in

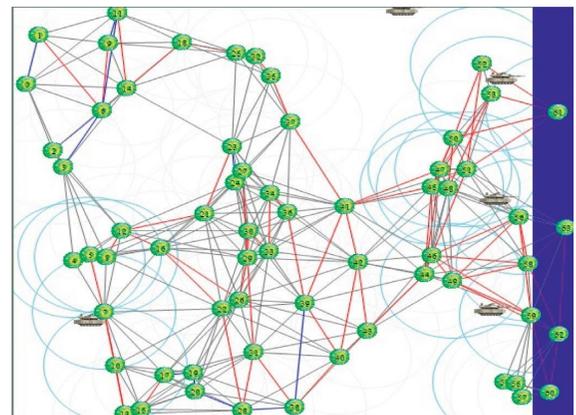


FIGURE 9: The operation of gossiping.

```

Gossiping (sender, receiver, root)
While (sender ≠ receiver and rootvalue ≠ null)
  New (rootlink) = sender;
  # We assume each node has a value link to the next and the neighbor.
  Sender = visitedneighbor;
  If sender == receiver
    Return true;
  Return true;
End while
Return true

```

ALGORITHM 2: Algorithm of Gossiping

```

Flood (Sender, receiver, Root)
Add Sender to queue
While (queue ≠ null)
  X = Remove (node(x)) from queue
  If X = receiver
    Return true
  Else
    Rootadd (X)
  For each nodes (neighbor X)
    If neighbor of X = null
      Remove neighbor X from the queue
  End while
Return true

```

ALGORITHM 3: Algorithm of Flooding

many sensor activities, such as transmitting the message, processing data, sensing, receiving. The amount of consumed energy can be calculated as shown below [21]:

$$\text{DataTransmission} : E_{tx}(d) = (\epsilon d^{2\alpha} + E_{elec})k, \quad (4)$$

where  $2 < \alpha \leq 4$

$$\text{Data reception} : E_{rx} = kE_{elec} \text{ represents,} \quad (5)$$

$$\text{Reception and Forward} : E_{total}(d) = (\epsilon d^{2\alpha} + 2E_{elec})k,$$

where  $E_{elec}$  represent the energy consumed in transmission,  $k$  is the size bits of message over a distance  $d$ ,  $\epsilon \in \{\epsilon_{fs}, \epsilon_{mp}\}$  is an amplifier of the transmitter in the free space ( $\sum fs$ ) or model of the multipath ( $\sum mp$ ), and  $\alpha$  is the path-loss exponent.

Therefore, the total power for each node is given as

$$E = E_{int} - E_{total}. \quad (6)$$

The leading causes of power loss and undesirable in the sensors network are the transfer of energy, collisions, hearing, listening to idle, and the control packages, over-processing nodes. The solutions to eliminate or reduce energy loss are choosing the suitable routing protocol, network design, nodes' synchronization, scheduling, and management or control capabilities implemented in each sensor [22].

Figures 10 and 11 illustrate how the framework evaluates power consumption. Figure 10 shows the monitoring

process at the beginning. This field displays the field power level, which indicates how much energy the sensors consumed, and the field status shows whether the sensors were active. The values decrease sequentially until they become zero. As illustrated in Figure 11, when the value equals zero, the sensor cannot perform any tasks (Death) since the amount of power is insufficient to accomplish any sensing, computation, or communication tasks.

## 5. Results Analysis

This section will discuss the results that appeared through the running of the simulation with our varying assumptions and parameters settings. We divided the problems into two parts: first, connectivity, coverage, and power consumption problems; second, deployment problem.

*5.1. Connectivity, Coverage, and Power Consumption Problems.* We divided these settings and assumptions into four stages. Parameter values were changed according to their impact on the power consumption and the sensor's operation, such as the range of communications and the sensor's radius, as well as the other assumption that is stated in each stage.

*Stage 1.* We assume that radius of the sensor's = 10, radius of the communication = 10, detection cost = 2 dB, sending cost = 5 dB, receiving cost = 2 dB, initial power = 1000 dB, and average time of each iteration (2 minutes).

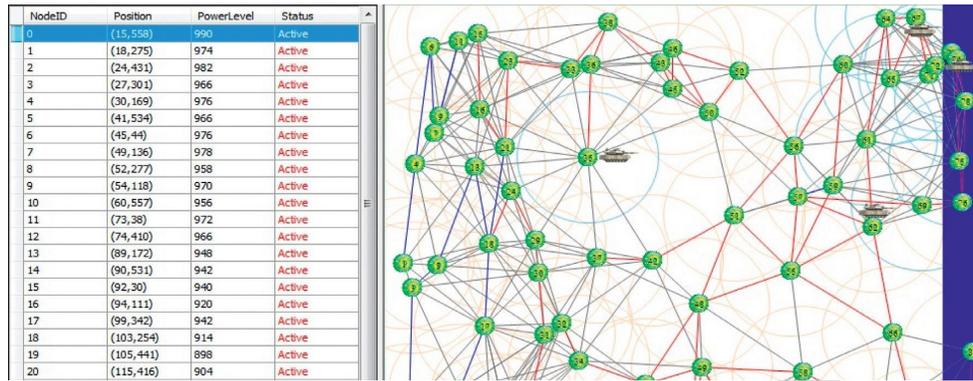


FIGURE 10: Example of power consumption 1.

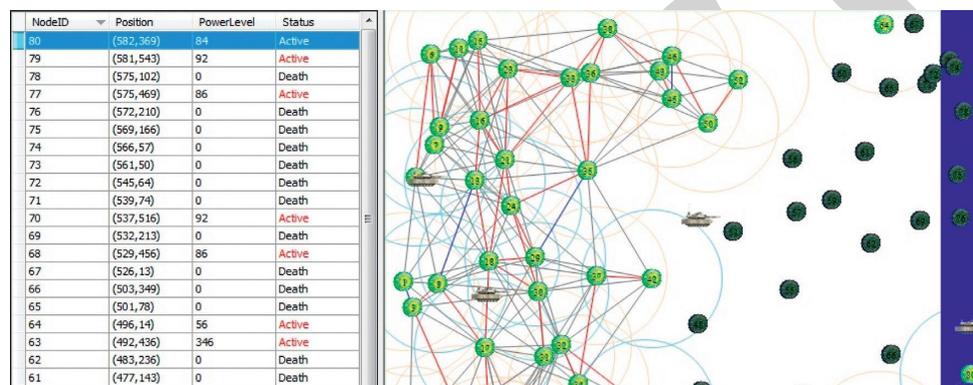


FIGURE 11: Power consumption 2.

Table 3 illustrates the problems of connectivity hole, coverage hole, and consumption of power in different interesting areas at a specific time, according to the assumptions mentioned in the first stage. We show that the connectivity problem is entirely nonexistent, which confirms the success of the proposed operation to solve the three problems facing the WSN. As for the coverage, the problem appeared when the area was 6 km, which is considered safe mode. On the other hand, energy consumption is relatively little except when the destination area is 2 km. In this case, energy consumption reaches 53%, in the worst case; also, this is a good result in the proposed method. Figure 12 also illustrates that.

*Stage 2.* We assume that the communication radius equal 10m, the radius of sensor's = 20m, detection cost = 2 dB, initial power = 1000 dB, sending cost = 5 dB, and receiving cost = 2 dB.

In the second stage, we observed an increase in power consumption when the radius of the sensor was increased, as well as a decrease in coverage, which had reached 11% to 15% because the number of sensors was reduced to less than half of what it was in the first stage with a radius of 20. This is illustrated in Table 4 and Figure 13. Therefore, that means the proposed model produces efficient results and confirms the success of the proposed operation to the three problems facing the WSN.

*Stage 3.* We assume that the radius of communication = 20, the radius of sensor's = 10, sending cost = 5 dB, detection cost = 2 dB, receiving cost = 2 dB, and initial power = 1000 dB.

Table 5 shows that when the communication radius of sensors is increased to 20, the coverage problem does not occur. The sensor radius was reduced by about 50% from Table 5(Stage 2). Consequently, the energy consumed is lower than that in Table 4(Stage 2). As sensors' communication radius increased, the connectivity problem disappeared. Figure 14 shows the proposed model produces efficient results and confirms the success of the proposed operation to the three problems facing the WSN.

*Stage 4.* We assume that the radius of communication = 20, radius of sensor's = 20, sending cost = 5 dB, detection cost = 2 dB, receiving cost = 2 dB, and initial power = 1000 dB.

In Table 6, we notice that increasing the sensor and communication radius led to a significant increase in power consumption. As a result, coverage problems and connectivity issues have since disappeared. In addition, Figure 15 shows the proposed model produces efficient results and confirms the success of the proposed operation to the three problems facing the WSN.

Table 7 shows a comparison between the four-stage results. As a result of our simulation, it was determined that a

TABLE 3: A connectivity problem, coverage problem, and power consumption (stage 1).

Area (K)	Net.Size	Conn.Problem	Cov.Problem	Avg.PC (%)
2	20	0	(1-5%)	53
4	40	0	(1-5%)	34
6	60	0	(6-10%)	35
8	80	0	(1-5%)	35
10	100	0	(1-5%)	34

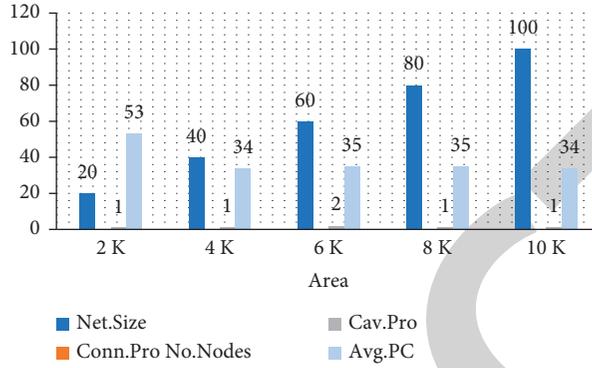


FIGURE 12: Representation of the data mentioned in stage 1.

TABLE 4: A connectivity problem, coverage problem, and power consumption (stage 2).

Area (K)	Net. Size	Conn.Problem	Co. Problem	Avg.PC (%)
2	5	0	(1-5%)	88
4	10	0	(6-10%)	65
6	15	0	(6-10%)	55
8	20	0	(11-15%)	55
10	25	1	(11-15%)	41

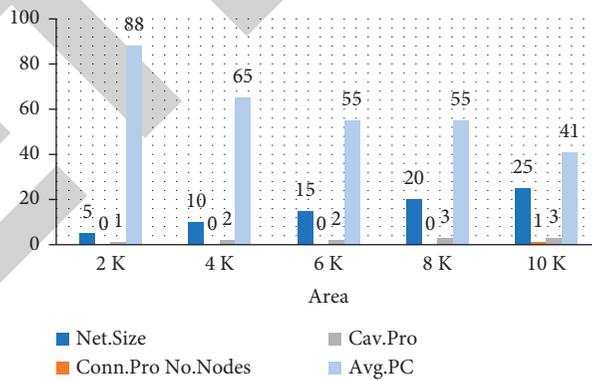


FIGURE 13: Representation of the data mentioned in Stage 2.

TABLE 5: A connectivity problem, coverage problem, and power consumption (Stage 3).

Area (K)	Net.Size	Conn.Problem	Cav.Problem	Avg.PC (%)
2	20	0	(1-5%)	46
4	40	0	(1-5%)	42
6	60	0	(1-5%)	39
8	80	0	(1-5%)	39
10	100	0	(1-5%)	37

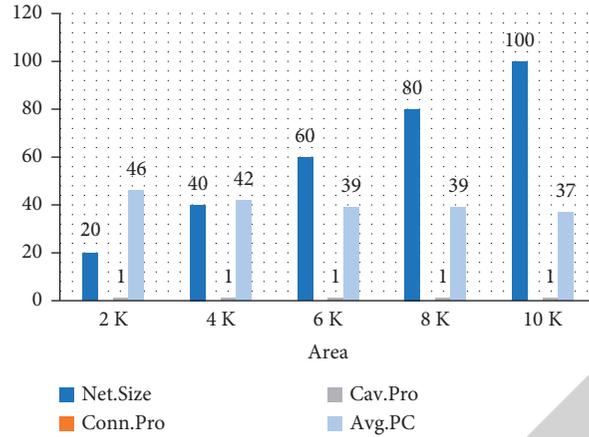


FIGURE 14: Representation of the data mentioned in Stage 3.

TABLE 6: A connectivity problem, coverage problem, and power consumption (Stage 4).

Area (K)	Net.Size	Conn.Problem	Cav.Problem	Avg.PC% (%)
2	5	0	(1-5%)	91
4	10	0	(1-5%)	72
6	15	0	(1-5%)	58
8	20	0	(1-5%)	47
10	25	0	(1-5%)	37

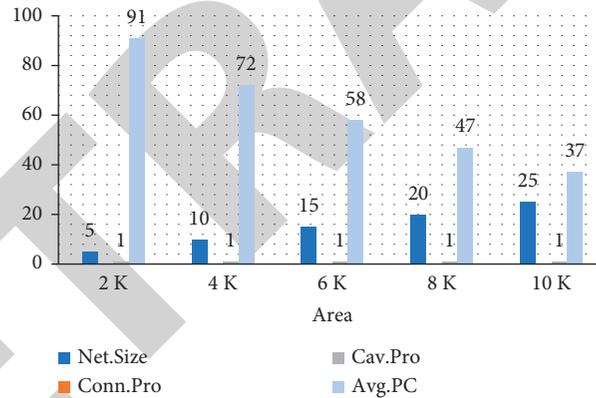


FIGURE 15: Representation of the data mentioned in Table 6.

TABLE 7: Comparison of the four-stage results.

Stage no.	Sensor's radius	Communication radius	Conn.Problem%	Cov. Problem	Power consumption (%)
Stage 1	10	10	(6-10%)	0	38
Stage 2	20	10	(1-5%)	1	61
Stage 3	10	20	(1-5%)	0	41
Stage 4	20	20	(1-5%)	0	61

radius of sensors = 10 and a communication radius = 20 provide the optimal settings for the sensors networks in Stage 3 and reduce connectivity and coverage hole problems and energy consumption.

According to the simulation results, the sensor closer to the uplink zone consumes a lot more power than those farther away. Since these sensors' death, they remain inactive in the target area. We developed an efficient algorithm in our

proposed framework, which led to less energy consumption in these sensors.

5.2. *Deployment Problem.* Figure 16, Figure 17, Figure 18, Figure 19, Figure 20, Figure 21, and Figure 22 show that different sensor numbers can be deployed based on the area and the sensor's radius. Sensors with a wide radius require fewer

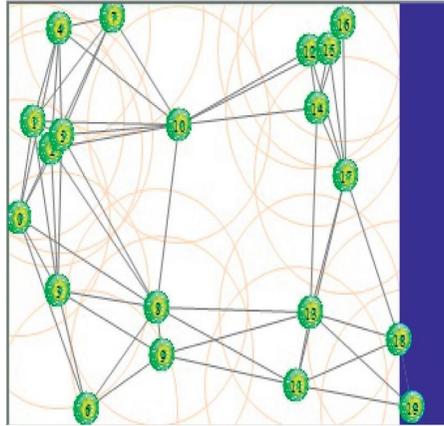


FIGURE 16: Sensor deployment with area = 2K, radius = 10, and NetSize = 20.

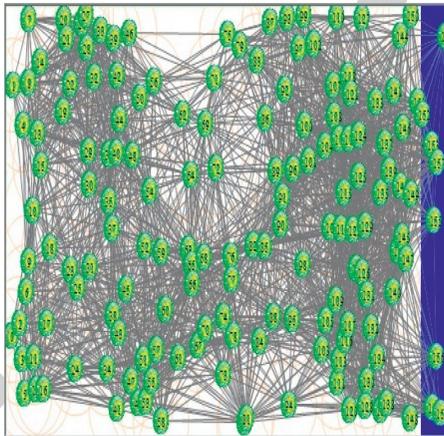


FIGURE 17: Sensor deployment with area = 2K, radius = 5, and NetSize = 80.

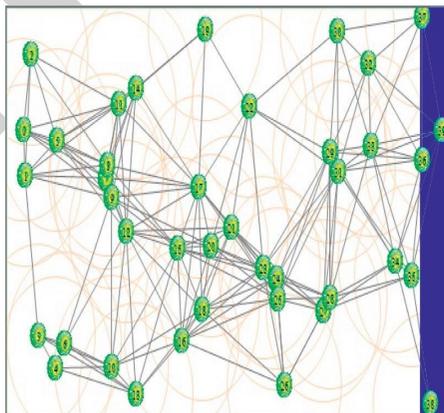


FIGURE 18: Sensor deployment with area = 4K, radius = 10, and NetSize = 40.

sensors, so coverage in a target area is better, reducing costs. Figure 16 shows, for example, a monitoring area of about 2 km around a sensor radius of 10m; 20 sensors are necessary to cover

the monitoring area. This same area is illustrated in Figure 17; however, the radius of the sensor is 5 m, which means that 80 sensors are needed, and so on for Figures 18 and 19.

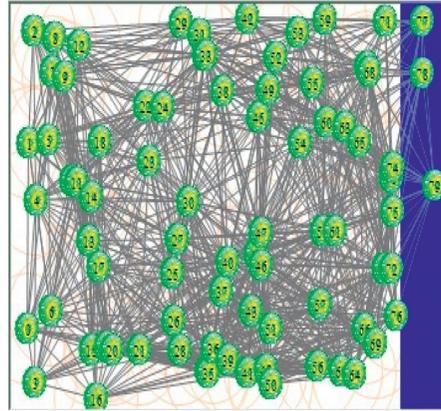


FIGURE 19: Sensor deployment with area = 4K, radius = 5, and NetSize = 80.

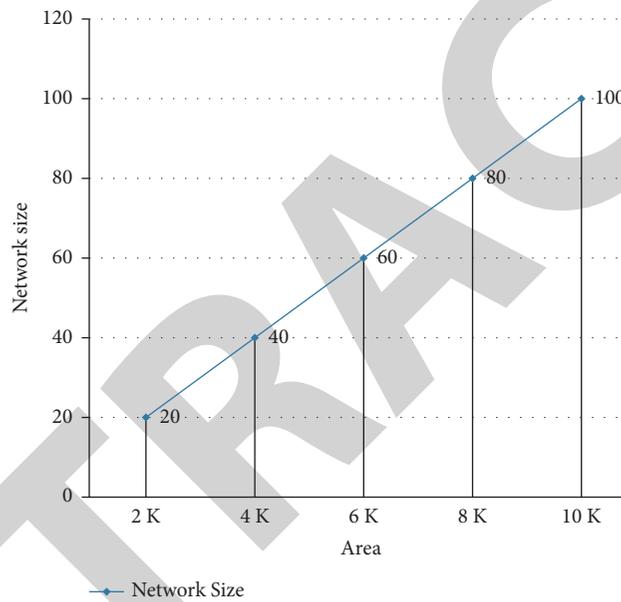


FIGURE 20: Sensors' deployment when the sensors' radius = 10 m.

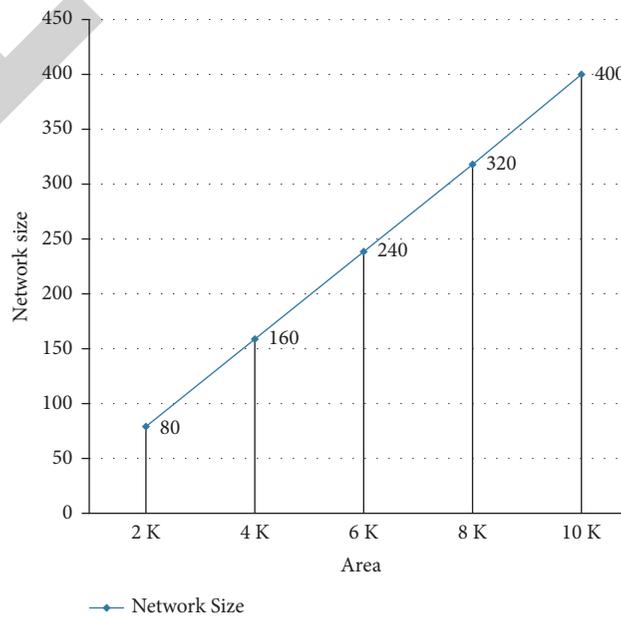


FIGURE 21: Sensors' deployment when the sensors' radius = 5 m.

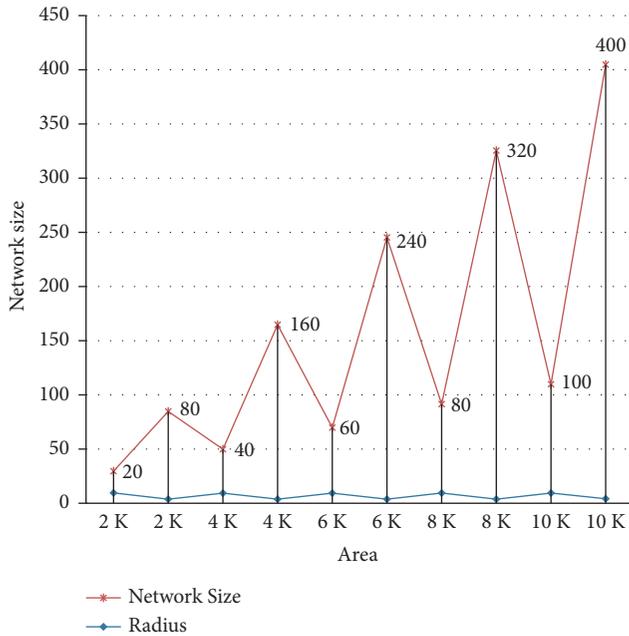


FIGURE 22: Sensors' deployment when the sensors' radius = 10 m and 5 m.

## 6. Conclusion

Wireless sensor networks (WSNs) are self-configuring, self-organized, infrastructure-free, and entirely dispersed networks. WSNs consist of heterogeneous/homogeneous sensors dispersed in a particular area to monitor phenomena such as pressure, pollution, sound, and temperature; then they transfer their data to a central location for analysis and decision-making. In this work, we combine WSN and MEC to produce a novel data collection framework (WSNEC). Therefore, we modelled the data collection process, including multiple sensors and tasks. As a result, we find an appropriate technique for deploying sensors in the interested area (e.g., military areas, isolated areas, and areas infested with terrorists) to cover the whole destination area based on set values for data quality. Finally, find an appropriate method for routing by selecting the routing protocol to improve their capabilities performance and decrease consumption of the energy to increase the life of the WSN. The framework which works as a simulation tool shows a good result and has been proven the efficient interaction between the MEC and WSN. We compare the results with traditional methods, which proved to outperform them.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by Project no, RSP-2021/206, King Saud University, Riyadh, Saudi Arabia.

## References

- [1] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [2] M. Alrazgan, "Internet of medical things and edge computing for improving healthcare in smart cities," *Mathematical Problems in Engineering*, vol. 2022, Article ID 5776954, 10 pages, 2022.
- [3] M. Al-Razgan, T. Alfakih, and M. M. Hassan, "A computational offloading method for edge server computing and resource allocation management," *Journal of Mathematics*, vol. 2021, no. 7, Article ID 3557059, pp. 1–11, 2021.
- [4] I. F. Akyildiz, W. Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [5] R. Andersson, M. Sandberg, and L. Urszuly, "Open secure office Project wireless sensor network," Tech.Rep. IDE0502, Halmstad University, Halmstad, Sweden, 2005.
- [6] D. J. Stein and Esq, "Wireless sensor network simulator v1.1," 2005.
- [7] H. K. Kalita and A. Kar, "Simulator based performance analysis of Wireless Sensor Network-A new approach," in *Proceedings of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 461–465, IEEE, Shanghai, China, November 2011.
- [8] T. Alfakih, M. M. Hassan, and M. Al-Razgan, "Multi-objective accelerated particle swarm optimization with dynamic programming technique for resource allocation in mobile edge computing," *IEEE Access*, vol. 9, pp. 167503–167520, 2021.
- [9] T. Alfakih, M. M. Hassan, A. Gumaie, C. Savaglio, and G. Fortino, "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA," *IEEE Access*, vol. 8, pp. 54074–54084, 2020.
- [10] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: a survey on architectures," *Infrastructure, and Algorithms*, vol. 52, no. 5, pp. 1–37, 2019.
- [11] J. Preden, J. Kaugerand, E. Suurjaak, S. Astapov, L. Motus, and R. Pahtma, "Data to decision: pushing situational information needs to the edge of the network," in *Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*, pp. 158–164, IEEE, Orlando, FL, USA, May 2015.
- [12] O. Debauche, S. Mahmoudi, and A. Guttadauria, "A new edge computing architecture for IoT and multimedia data management," *Information*, vol. 13, no. 2, p. 89, 2022.
- [13] G. Liu, Z. Liu, V. S. Sheng, L. Zhang, Y. J. W. C. Yang, and M. Computing, "A novel energy-efficient, static scenario-oriented routing method of wireless sensor network based on edge computing," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 3450361, 2022.
- [14] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobilead-hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 6, pp. 387–395, 2001.
- [15] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, Wiley-IEEE Press, New York, United States, 2009.

- [16] S. Deng, L. Shen, and J. Li, "Mobility-based clustering protocol for wireless sensor networks with mobile nodes," *IET Wireless Sensor Systems*, vol. 1, no. 1, pp. 39–47, 2011.
- [17] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks," in *Proceedings of the First International Conference on Wireless Internet (WICON'05)*, pp. 114–121, IEEE, Budapest, Hungary, September 2005.
- [18] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor networks: a survey," *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.
- [19] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pp. 80–91, ACM, Washington, DC, USA, June 2002.
- [20] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [22] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the ICC 2001. IEEE International Conference on Communications*, vol. 2, pp. 472–476, IEEE, Helsinki, Finland, August 2001.