

Dynamic Weather-Adaptive Enhanced Barrier Coverage with Adjustable-Range Sensors for WRSNs

Pei Xu¹, Youxi Li², Chih-Yung Chang³, Chin-Hwa Kuo^{3*}, Diptendu Sinha Roy⁴

¹ School of Artificial Intelligence and Big Data, Hefei University, China

² Hangzhou Ruiyuan Jiling Technology Co. Ltd, China

³ Department of Computer Science and Information Engineering, Tamkang University, Taiwan

⁴ Department of Computer Science and Engineering, National Institute of Technology Meghalaya, India
 xupei@hfu.edu.cn, liyouxi2010@gmail.com, cychang@mail.tku.edu.tw,
 chinhwakuo@mail.tku.edu.tw, diptendu.sr@nitm.ac.in

Abstract

Barrier coverage is vital for wireless sensor networks (*WSNs*). Traditional approaches using battery-powered, fixed-radius sensors under the Boolean Sensing Model (*BSM*) struggle to ensure long-term, high-quality monitoring. This paper proposes *BCRAS*, a barrier coverage algorithm based on solar-powered sensors with adjustable sensing radii and the Probabilistic Sensing Model (*PSM*). It addresses three key challenges: (1) To cope with solar power uncertainty, a CNN-LSTM model predicts next-day *PV* energy to support energy-aware scheduling; (2) To manage varying energy consumption across sensing ranges, each sensor selects its sensing radius based on predicted energy gain and usage balance; (3) To enhance coverage under *PSM*, sensors are scheduled according to their cooperative detection probability at bottleneck points. Experiments show that *BCRAS* improves surveillance quality, energy utilization, and long-term stability compared to existing methods.

Keywords: Barrier coverage, Wireless rechargeable sensor networks, Range-adjustable sensor, Probabilistic sensing model

1 Introduction

Wireless sensor networks have been applied to a wide range of applications including earthquake detection and prediction, environment monitoring, and intruder detection in hazardous regions [1-2] *et al.* The coverage problem is one of the most important issues in *WSNs*, which can be classified into three categories: target coverage, area coverage, and barrier coverage. The target coverage aims to monitor some specific objects [3] such as museums and campuses while the area coverage aims to monitor a given region [4] such as agricultural test fields and chemical plants. Unlike target and area coverages, barrier coverage aims to detectability of intruders when they are crossing the border of an important surveillance area, it has been widely employed to prevent unauthorized invasion [5].

The barrier coverage problem has been widely discussed in the literature. The main challenge of the barrier coverage is the limited detection capability and limited battery. Consequently, achieving high surveillance quality while maintaining a long network lifetime has been the common goal of existing studies [6-7]. However, the detection capability of sensors is limited, and replacing the battery of sensors is impractical in many scenarios.

The surveillance quality of the barrier is generally determined by the sensing model and the barrier construction algorithm. Most of the existing studies designed barrier construction algorithms based on the Boolean Sensing Model (*BSM*) which aimed to construct a barrier that satisfies *k*-barrier coverage [8, 22]. The *BSM* assumes that the sensing ability of each sensor is perfect and the intruder located within the sensing radii of each sensor is guaranteed to be detected. Under this assumption, the *BSM* is unable to reflect the physical features of sensing. Different from *BSM*, the Probability Sensing Model (*PSM*) describes the sensing ability by detection probability [9]. The algorithms based on *PSM* addressed the barrier coverage problem and aimed to detect intruders with a probability larger than the predefined requirement. The *PSM* is more practical as compared with the *BSM* because it matches real sensing behavior. However, most of the previous studies adopted the sensor with a fixed sensing radius [9-11]. According to the *PSM*, the detection capability of a sensor increases as its sensing radius expands. Additionally, it is observed that a fixed sensing radius may not be flexible enough to fully utilize the detection capability of each sensor.

In recent years, the use of sensors with adjustable sensing radii has become more prevalent [12-14]. This presents new challenges for sensor scheduling algorithms. With adjustable sensing radii, the scheduling task for each sensor becomes more complex. Additionally, sensors with different sensing radii consume varying amounts of energy over the same period, leading to increased complexity in energy allocation. Although some computational intelligence methods, such as multi-objective optimization [27], can effectively solve such problems, most devices in *WSNs* have limited computational capabilities, making them unable to use such good methods.

*Corresponding Author: Chin-Hwa Kuo; Email: chinhwakuo@mail.tku.edu.tw

The network lifetime is another challenge of the barrier coverage problem. Some studies assumed that the *WSNs* are battery-powered and designed the optimal sleep and wakeup algorithm [15, 23], or the energy-efficient algorithm to prolong the network lifetime [16-17]. However, the performance of *WSNs* is still constrained by their limited energy. To improve the performance of *WSNs*, rechargeable sensor networks (*WRSNs*) have been widely applied, and plenty of energy-replenishing algorithms have been proposed. These algorithms can be classified into two types, including mobile charger charging [18-19], and environmental energy harvesting [14, 20]. Although mobile chargers can provide stable and sufficient energy, however, barrier coverage is usually applied in large outdoor areas. Besides, in some application scenarios, the terrain can present considerable complexity. The cost of large-scale deployment of mobile chargers is substantial. Therefore, environmental energy harvesting is more suitable for barrier coverage.

The environmental energy harvesting technology primarily derives energy from wind and solar sources. Compared to solar energy, wind energy is more unstable, and the locations with sufficient wind resources are fewer. Most existing studies have overlooked the impact of different weather conditions on solar energy while assuming the photovoltaic (*PV*) power to be a constant value. However, the *PV* power is variable and dependent on weather conditions. Consequently, the evaluation of *PV* power under this assumption will be conservative and rough, which limits the performance of *WRSNs*.

As discussed above, the use of solar-powered sensors with adjustable sensing radii introduces several new challenges: (1) Solar-power is a dynamic energy resource and is affected by weather conditions, making the scheduling of sensor operations more complex; (2) Sensors with different sensing radii consume varying amounts of power, complicating energy allocation due to the multiple possible schedules. (3) The cooperative detection among sensors should be considered, as the monitoring contribution of each sensor differs depending on whether it operates independently or in cooperation with neighboring sensors when *PSM* is applied. Despite these pioneering efforts, an effective and efficient algorithm for barrier coverage to maximize the surveillance quality while maintaining the perpetual network lifetime is still lacking.

To address these challenges and achieve the objectives of maximizing the surveillance quality while balancing the acquired and consumed energy of each sensor each day, this paper proposes a *Barrier Coverage algorithm for Range Adjustable Sensors in WRSNs*, called *BCRAS* in short. To address the first challenge, the CNN-LSTM model is initially employed to predict the photovoltaic (*PV*) power function of the next day in advance, so that the available solar energy of the next day can be predicted accurately. Subsequently, to address the second challenge, the *BCRAS* partitions the time into several identical cycles, each consisting of a fixed number of identical time slots. Based on the predicted available solar energy, *BCRAS* evenly distributes the available energy across all cycles. The active time of each cycle is determined

by the associated sensing radius. To ensure consistent surveillance quality, each sensor follows the same task schedule in every cycle. Finally, to address the third challenge, the *BCRAS* partitions the monitoring region into several identical grids, and the detection probability of each sensor to all points within the same grid is considered to be identical. These grids and time slots form multiple space-time points. The calculation of the cooperative detection probability of active sensors to each space-time point is straightforward. Additionally, this partition scheme enables the calculation of surveillance quality across both spatial and temporal dimensions. The *BCRAS* first identifies the space-time point with the weakest detection probability, referred to as the bottleneck. It then prioritizes the scheduling of the sensor that provides the highest surveillance contribution among its neighboring sensors to cover this bottleneck space-time point. By leveraging this strategy, cooperative detection improves the overall monitoring quality.

The main contributions of the proposed *BCRAS* algorithm are itemized as follows:

(1) Considering the variation of *PV* Power. Most of the existing studies [16-17] assumed that *PV* power is constant. However, the *PV* power varies over time and is influenced by weather conditions. Because *PV* power data is generated sequentially over time, it exhibits temporal correlation. Additionally, *PV* power data is influenced by weather factors, resulting in significant spatial correlation. The proposed *BCRAS* utilizes the CNN-LSTM model to extract both time-series features and spatial features from historical meteorological data and *PV* power data, to accurately predict the available solar energy of the next day.

This contribution addresses the challenge of unpredictable and weather-dependent solar energy availability, which complicates energy-aware sensor scheduling.

(2) Improving the utilization of solar power while balancing the consumed and acquired energy of each sensor. Based on the predicted solar power for the next day, the *BCRAS* algorithm allocates energy for sensor operation during both daytime and nighttime, preventing sensors from depleting more energy than they acquire, which could lead to failure. This precise energy management avoids an overly conservative approach, enhancing solar energy utilization. Additionally, each sensor calculates its active time under different sensing radii based on the allocated energy, ensuring a daily balance between energy gained and consumed, thereby extending the network lifetime. This contribution addresses the challenge of energy allocation complexity caused by varying energy consumption at different sensing radii.

(3) Developing the contribution-based strategy for cooperative sensing. Considering the *PSM*, the *BCRAS* algorithm recognizes the differing contributions of each sensor, whether they are operating cooperatively or independently. When multiple neighboring sensors are set to activate simultaneously, they are considered as a collective unit. The *BCRAS* calculates the

cooperative detection probability of each sensor to every space-time point and identifies the point with the weakest cooperative detection probability. Then, the sensor with the maximum surveillance contribution is scheduled to monitor this bottleneck point, to improve the surveillance quality of the area based on the cooperative detection between neighboring sensors.

This contribution addresses the challenge of scheduling under the probabilistic sensing model, where cooperative detection must be optimized to maximize surveillance.

(4) Enhancing the surveillance quality by adjusting the sensing radius. Since the sensors are randomly deployed, the density of the *WRSNs* varies in different areas. Therefore, the areas with high-density deployed sensors might have redundant detections. The proposed *BCRAS* algorithm constructs the task schedule of each sensor to dynamically adjust the sensing radius and active time slot, aiming to utilize the potential for redundant detection capabilities. As a result, the surveillance quality achieved by the proposed *BCRAS* algorithm surpasses that of existing algorithms that adopted fixed sensing radius sensors.

This contribution addresses the challenge of redundant sensing in high-density areas and enables fine-grained control of detection resources.

The remainder of this paper is organized as follows. Section 2 summarizes the related work. Section 3 formalizes the assumptions and problem statement of the investigated issue. In Section 4, the *BCRAS* algorithm is introduced. Section 5 presents the performance study and Section 6 gives conclusions and future work of this paper.

2 Related Work

In literature, several barrier coverage algorithms have been proposed in recent years. In general, these studies can be partitioned into four categories: *WSNs* with Fixed Sensing Radius Sensors, *WSNs* with Adjustable Sensing Radius Sensors, *WSNs* with Battery-powered Sensors, and *WSNs* with Rechargeable Sensors. The following reviews these studies and compares them with our work.

2.1 WSNs with Fixed Sensing Radius Sensors

Because the fixed sensing radius sensor is comparatively easier to analyze and design the coverage algorithm, several studies applied the fixed sensing radius to investigate the barrier coverage issue. Study [22] proposed an autonomous deployment algorithm (*MobiBar*) for k -barrier coverage in mobile sensor networks. The mechanism coordinated the sensor movements and constructed k distinct complete barriers to ensure the desired level of redundancy which achieves the maximum level of barrier coverage. To further enhance the robustness of the barrier coverage, study [10] presented a coverage-aware distributed k -connectivity maintenance algorithm that generated minimum-cost movements of active nodes after a node failure to preserve a persistent k value subject

to a coverage conservation criterion. The algorithm accepted a coverage conservation ratio as a trade-off parameter between coverage and movements and facilitated the coverage with the generated movements according to this value. It can efficiently restore the k -connectivity. Both of the above-mentioned studies adopted the *BSM*, which might not be accurate in real applications. Some other studies adopted the *PSM* to design the barrier coverage algorithm.

Study [9] investigated the efficient utilization of mobile sensors with the *PSM* to enhance barrier coverage. Initially, they defined the concept of a “safe cell” and developed algorithms to identify safe cells and barrier gaps. Subsequently, they devised an algorithm aimed at optimizing barrier coverage using the fewest mobile sensors. In the final step, the Kuhn-Munkres (*KM*) algorithm was employed to address the problem of minimizing mobile sensor movement. To achieve qualified barrier coverage, Study [11] addressed the barrier coverage problem and applied the game theory to account for various paths that the intruders may take. They proposed an iterative algorithm to refine the set of candidate defender formations. Given the set of candidate formations, a mixed Nash equilibrium gives a stochastic policy to deploy the defenders. Study [31] proposed a hybrid algorithm combining dynamic multi-swarm PSO and firefly algorithm to address coverage hole problems in *WSNs*. By introducing a sub-swarm switch mechanism and Lévy flight, the method enhances search diversity and avoids premature convergence, achieving improved coverage and reduced energy consumption. Though the sensor with a fixed sensing radius is easy to analyze, the cooperative sensing opportunities by adjusting the sensing radius still need to be further explored.

2.2 WSNs with Adjustable Sensing Radius Sensors

To further improve the performance of *WSNs*, some studies adopted sensors with adjustable sensing radii. Study [12] investigated the barrier coverage problem in Directional Sensor Networks (*DSNs*), considering the nodes’ capability to adjust their working directions and sensing radii. This paper proposed a barrier construction scheme that schedules nodes to collaboratively form multiple barriers, determining their optimal working directions and sensing radii. This scheme facilitates the formation of additional barriers, thereby extending the service lifetime of the network. However, directional sensors, such as cameras, represent a specialized category of sensors. As they are not universally prevalent, the scheme proposed in this paper may not be suitable for the majority of scenarios.

Study [13] addressed the coverage problem of *WSNs* by considering the range-adjustable sensors. Based on the *BSM*, they proposed a neighborhood-based estimation of distribution algorithm (*NEDA*) to address it recursively. In *NEDA*, each individual represented a coverage scheme in which the sensors were selectively activated to monitor all the targets. A linear programming (*LP*) model was built to assign activation time so that the network lifetime can be maximized while the surveillance quality can be

guaranteed. However, employing the *NEAD* algorithm to identify the global optimal solution for the coverage problem is highly complex and entails significant time and energy consumption. Study [14] proposed an algorithm that utilizes sensors with adjustable sensing radii. Initially, all sensors are scheduled to operate at their maximum sensing radius. Subsequently, the algorithm reduces the sensing radius of some sensors to further improve the detection probability at the weakest points of interest. However, this approach may not be sufficiently flexible to determine the optimal sensing radius for each sensor, potentially limiting the overall efficiency of the *WSNs*.

2.3 WSNs with Battery-powered Sensor

For the *WSNs* with battery-powered sensors, the energy is limited, therefore, most of the studies aim to develop energy conservation or energy-efficient schemes to extend the network lifetime. Study [23] addressed minimizing the number of sensors in a 3-D wireless sensor network for k -coverage of a field of interest. A probabilistic approach estimated redundancy, and a distributed protocol scheduled redundant sensors to sleep, reducing active sensors and extending network lifetime. Study [15] studied the problem of finding a lifetime-maximizing subset with a breach-free sleep-wakeup scheduling to achieve k coverage. They first proved that determining whether a given sleep-wakeup schedule is breach-free can be done in polynomial time. They then demonstrated that the problem of finding a lifetime-maximizing, breach-free schedule is equivalent to the maximum node-weighted path problem in a directed graph and designed a parameterized algorithm. Study [16] presented a two-phase approach for sensor placement and scheduling in *WSNs*. In the first phase, a novel heuristic, Greedy Intersecting Arc Selection, was used to maximize k coverage, while in the second phase, a Memetic Set Scheduler optimized the network lifetime. Experimental results demonstrated that the proposed method outperformed existing approaches in both solution quality and execution time. However, these studies adopted the *BSM*, which may not be accurate.

Study [17] proposed a secure and energy-efficient barrier coverage algorithm, called *SEC²*. The *SEC²* used a fully weighted attributed dynamic graph model in which a novel attribute-based weight-balancing greedy strategy was used to construct barriers, aiming to guarantee the Quality of Service (QoS) at all times in an energy-efficient way. However, the solution obtained through a greedy strategy may not be sufficiently effective, or could even be bad in the globe. Although the above studies proposed various methods to extend the lifetime of wireless sensor networks, energy constraints continue to limit WSN performance.

2.4 WSNs with Rechargeable Sensors

To overcome the energy constraint, some studies adopted the mobile charger to periodically visit the sensors and recharge them. Study [18] proposed the *PLPR* algorithm for energy recharging in *WRSNs*, focusing on minimizing the number of mobile chargers (*MCs*) while

ensuring continuous sensor operation. The algorithm optimizes charging paths and introduces cooperation between *MCs* and super *MCs* to enhance energy efficiency. Simulation results show that *PLPR* outperforms existing methods in reducing energy consumption and recharging time. However, the algorithm still faces challenges in large-scale deployments, particularly in balancing energy consumption across all sensors.

By dynamically adjusting the recharging schedule based on the received recharging requests from sensors in real-time, Study [19] proposed a reinforcement learning approach, called reinforcement learning recharging, for mobile chargers to learn the trends of *WSNs*, including the energy consumption of the sensors, the recharging cost as well as the coverage benefit, aimed to maximize the coverage contribution of the recharged *WSN*. Study [32] introduced a contribution-aware charging mechanism that partitions the monitoring area into hexagonal grids and schedules charging based on sensor importance to surveillance quality and connectivity. This approach improves the effectiveness of mobile recharging in maintaining network performance. Study [33] introduced the *MC3* (Multi charger Cooperative Charging) mechanism, which employs multiple mobile chargers (*MCs*) to optimize energy replenishment in wireless rechargeable sensor networks (*WRSNs*). *MC3* dynamically adjusts recharging regions to minimize waiting times and energy consumption, while balancing the workload of each *MC*. This approach improves surveillance quality and recharging efficiency, outperforming existing algorithms in terms of coverage and effectiveness. However, in some application scenarios of the barrier coverage, the *WSNs* were deployed in a wide range of wild areas. It is impractical to visit each sensor by mobile charger individually. Some environmental energy harvesting (*EH*) technologies were developed for barrier coverage.

The environment energy mainly includes solar and wind. Compared to wind energy, solar energy is more abundant and common. Study [20] proposed a maximizing cooperative detection probability (*MCDP*) algorithm, which scheduled each sensor to be periodically activated and recharged. It scheduled the sensors to maximize the detection probability of every point in the monitoring region. The *MCDP* aimed to maximize the surveillance quality of the constructed barrier while perpetuating the network lifetime. However, the *MCDP* algorithm still utilized sensors with a fixed sensing radius. Study [14] introduced a target coverage mechanism, *TCSAR*, which aimed to enhance surveillance quality at points of interest (*POIs*) and extend network lifetime by coordinating the activation of solar-powered sensors. *TCSAR* aimed to select the optimal scheduling for each *POI* and adjust sensors' sensing radii to improve surveillance. However, this paper assumed that *PV* power is constant. Moreover, both studies [14, 20] didn't consider the impact of different meteorological scenarios on *PV* power.

Table 1 gives a comparison of the main characteristics of the proposed *BCRAS* algorithm and the existing works.

Table 1. Comparisons of the main characteristics of the proposed *BCRAS* with the existing related work

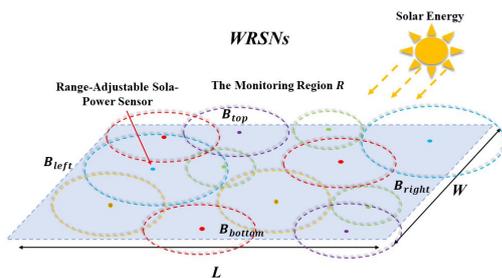
Related work	<i>PSM</i>	Adjustable sensing radius sensor	Rechargeable sensor	Considering the impact of meteorological scenarios on <i>PV</i> power	Maximizing the surveillance quality
[9]	○	×	×	×	×
[10]	×	×	×	×	×
[11]	○	×	×	×	○
[12]	○	○	×	×	○
[13]	×	○	×	×	×
[14]	○	○	○	×	○
[15]	×	×	×	×	×
[16]	×	×	×	×	×
[17]	×	×	×	×	×
[18]	×	×	○	×	×
[19]	×	×	○	×	×
[20]	○	×	○	×	○
[22]	×	×	×	×	○
[23]	×	×	×	×	×
[31]	×	×	×	×	×
[32]	×	×	○	×	×
[33]	×	×	○	×	×
<i>BCRAS</i>	○	○	○	○	○

3 Assumption and Problem Statement

This section initially presents the network model and assumptions of the considered *WRSNs*. Then, the objective and constraints of the investigated problem are described.

3.1 Network Environment

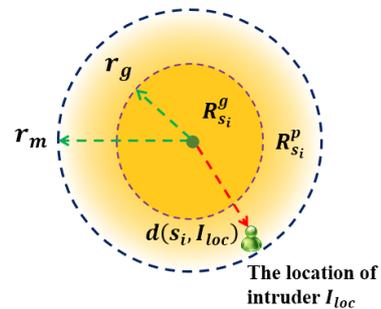
This paper considers a rectangle monitoring region R with dimensions $L \times W$, where L and W are the length and width of R , respectively. Let B_{left} , B_{right} , B_{top} and B_{bottom} denote the left, right, top, and bottom boundaries of R . Let (x_v, y_v) represent the coordinates of a given point v in R . We assume that a set of solar-powered sensors $S = \{s_1, s_2, \dots, s_n\}$ have been randomly deployed in region R . The sensing radius of each sensor is identical and adjustable with several fixed levels. The power consumption rate and detection capability of each sensor will be increased with its sensing radius. Each sensor has a unique *ID* and is aware of its location, remaining energy and the boundaries of R . Let (x_i, y_i) represent the coordinate of the deployed sensor s_i . The communication radius is assumed at least twice the maximum sensing radius. Figure 1 gives the scenario of the considered network.


Figure 1. An example of considered *WRSNs*

3.2 Sensing Model

The *PSM* is applied in this paper. Generally, a sensor is more likely to detect an intruder when the target is closer to it. Figure 2 illustrates the logical view of the *PSM*.

Let $\mathbb{R} = \{r_1, r_2, \dots, r_m\}$ denote the set of selectable sensing radiuses of each sensor, where $r_i < r_j$ if $i < j$. As shown in Figure 2, the sensing radii of s_i can be divided into two regions. The inner region is guaranteed sensing region $R_{s_i}^g$, and the outer region is the probabilistic sensing region $R_{s_i}^p$, with radii r_g and r_m , respectively. If an intruder is located within $R_{s_i}^g$, sensor s_i detects the intruder with 100% probability. However, if the intruder is within $R_{s_i}^p$, the detection probability of s_i to intruder decreases with increasing distance from the sensor, ranging from 100% to 0%.


Figure 2. The probabilistic sensing model

Let I_{loc} denote the location of the intruder. Let $p(s_i, I_{loc})$ denote the detection probability of s_i to point intruder at the location I_{loc} . Let $d(s_i, I_{loc})$ denote the Euclidean distance between the s_i and I_{loc} . The value of $p(s_i, I_{loc})$ can be calculated by applying the following expression.

$$p(s_i, I_{loc}) = \begin{cases} 1 & \text{if } d(s_i, I_{loc}) \leq r_g \\ e^{-\lambda \alpha^\gamma} & \text{if } r_g < d(s_i, I_{loc}) \leq r_m \\ 0 & \text{if } d(s_i, I_{loc}) > r_m \end{cases} \quad (1)$$

where $\alpha = d(s_i, I_{loc}) - r_g$, λ and γ represent the path loss exponents of the sensing signal strength of the sensor.

In the studied *WRSNs*, path loss plays a pivotal role in determining the overall effectiveness of the proposed algorithm. Path loss refers to the reduction in signal energy as it propagates through the environment, directly influencing the communication range and signal quality between sensors. An increase in path loss necessitates higher transmission power or more frequent communication attempts by the sensors, both of which contribute to elevated energy consumption [29]. As a result, the energy available for sensing operations diminishes, ultimately degrading the monitoring quality.

The *PSM* utilized in the *BCRAS* algorithm is governed by Exp. (1), where parameters λ and γ significantly influence the path loss attenuation model. The parameter λ dictates the rate at which signal strength decreases over distance, while γ modulates the shape of the attenuation function, further refining the dynamic range of signal decay.

To comprehensively analyze the effect of path loss on coverage performance within the examined *WRSNs*, this study conducts an in-depth evaluation of its influence on the overall performance of the *BCRAS* algorithm based on the simulation experiments presented in Section 5.

3.3 Charging and Discharging Model

In the considered *WRSNs*, all sensors are solar-powered. During the daytime, each sensor can be in one of two states: sensing & charging or charging-only. It is noted that solar power is not available at nighttime. Hence, the possible states for each sensor are either sensing-only or sleeping during nighttime. Energy is consumed by each sensor when it is in the sensing & charging state or the sensing-only state. Sensors in these states are referred to as active sensors. Sensors can be charged while in the sensing & charging state or charging-only state. In the sensing & charging state, each sensor can perform sensing and charging operations simultaneously. Conversely, a sensor in the sleeping state does not perform any operation or consume energy.

Let E denote the battery capacity of each sensor. Let $\mathcal{P}_{i,k}^{sen}$ denote the power consumption rate when the sensor s_i selects the sensing radius $r_k \in \mathbb{R}$. According to the study [24], the power consumption rate of the sensor s_i corresponds to its sensing radius and can be quantified by the following expression.

$$\mathcal{P}_{i,k}^{sen} = \mathcal{P}_{i,m}^{sen} \times (r_k / r_m)^2 \quad (2)$$

where $\mathcal{P}_{i,m}^{sen}$, the power consumption rate of the largest sensing radius r_m , is given in advance.

The proposed *BCRAS* algorithm partitions the time axis

into several equal-length cycles, each cycle comprising numerous identical time slots. This partitioning facilitates consideration of surveillance quality in both spatial and temporal dimensions. Given that energy consumption peaks when a sensor uses its maximum sensing radius r_m , it is assumed that the energy allocated for each cycle is only sufficient to sustain the sensor being activated at r_m for one time slot per cycle. This time slot, denoted by τ , represents the basic time unit in each cycle for task scheduling. The length of one time slot and one cycle will be discussed in section 4.

For ease of calculation, assume that the relation between any sensing radius r_k and the maximum sensing radius r_m is given by the following expression

$$\frac{r_m}{r_k} = \sqrt{m-k+1} \quad (3)$$

According to Exps. (2) and (3), when s_i selects sensing radius r_m , the active time of the sensor s_i , denoted by $t_{i,k}^{sen}$, within one cycle is given by:

$$t_{i,k}^{sen} = (m-k+1) \cdot \tau \quad (4)$$

Figure 3 gives an example of the considered model. In this example. We have $\mathbb{R} = \{r_1, r_2, r_m\}$. According to Exp. (3), we have

$$r_k = \frac{r_m}{\sqrt{m-k+1}}, \quad r_1 = \frac{r_m}{\sqrt{m}}$$

As shown in Figure 3, the sensor s_i can be activated in one time slot when it adopts the maximum sensing radius r_m . The active time of s_i selecting sensing radiuses r_k and r_1 are $(m-k+1) \cdot \tau$ and $m \cdot \tau$, respectively.

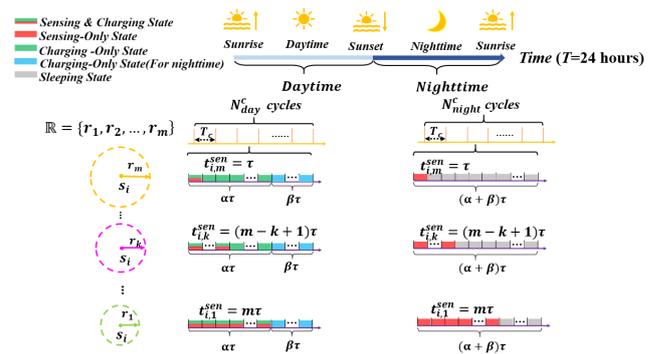


Figure 3. An example of charging and discharging model

3.4 Transmission Scheduling Protocol

In the investigated Wireless Rechargeable Sensor Networks (*WRSNs*), the communication channel is modeled as a quasi-static fading channel. This assumption is based on the fact that the sensors used in this study are stationary and have predetermined deployment locations, resulting in gradual variations in channel conditions. Consequently, the channel remains stable over an extended

period, with significant fluctuations occurring only over longer durations. The quasi-static fading channel model more accurately reflects real-world channel characteristics, thereby enhancing the practical applicability of the proposed algorithm.

Under this assumption, the carrier sense multiple access with collision avoidance (CSMA/CA) protocol is employed to enhance communication efficiency and mitigate data collisions. The primary rationale for this choice is that in a quasi-static fading channel, short-term stability facilitates the effectiveness of the CSMA/CA collision avoidance mechanism. Furthermore, the implementation of CSMA/CA is relatively straightforward, requiring sensor nodes to perform channel sensing and simple random backoff computations, which imposes minimal computational overhead. Additionally, by minimizing packet collisions and retransmissions, CSMA/CA effectively reduces energy consumption. Compared to TDMA and CDMA, CSMA/CA does not require global clock synchronization or complex code allocation, making it more adaptable to distributed network scenarios. Given that the proposed JMLA algorithm operates in a distributed manner, CSMA/CA's decentralized functionality aligns well with the application requirements of this study.

Let Boolean variables $c(t)$ represent whether the channel state is idle or busy at time t . That is

$$c(t) = \begin{cases} 1, & \text{if the channel state is idle at time } t \\ 0, & \text{if the channel state is busy at time } t \end{cases}$$

Each sensor s_i initially checks the current channel state $c(t)$ before attempting to transmit data. If the channel is idle ($c(t) = 1$), s_i proceeds to the transmission preparation phase. If the channel is busy ($c(t) = 0$), s_i enters the backoff phase. Let T_i^w denote the random backoff waiting time for the sensor s_i when $c(t) = 0$. In the backoff phase, the sensor s_i initially calculates the backoff time T_i^w using an exponential backoff algorithm controlled by the backoff stage k , that is.

$$T_i^w = \text{Random}(0, 2^k - 1) \times B_{\text{unit}}$$

where B_{unit} is the backoff time unit, which represents the fundamental unit of time that a network node must wait before attempting to retransmit data after detecting that the channel is busy.

After the backoff time ends, s_i rechecks the channel status. Upon confirming that the channel is idle, it sends a Request to Send (RTS) signal, requesting data transmission. When the receiver receives the RTS frame, it checks if the channel is currently idle and if it is ready to receive data. If both conditions are met, it sends a Clear to Send (CTS) frame back to the s_i , indicating that it is clear to send. transmission. After receiving the CTS response, s_i begins data transmission operation. Once the transmission is completed, s_i receives an acknowledgment signal (ACK), confirming that the data has successfully reached the receiver.

3.5 Problem Statement

This paper addresses the barrier coverage problem in WRSNs and utilizes sensors with adjustable sensing radii. The proposed BCRAS algorithm aims to design the best task schedule for each sensor in a distributed manner. It aims to maximize the surveillance quality while balancing the acquired and consumed energy. The objective function and associated constraints are presented as follows.

Let \mathbb{s} denote a possible scheduling algorithm. Let $DB_{\mathbb{s}}$ denote a constructed barrier based on \mathbb{s} . Let t_h denote the h -th time slot in the cycle T_c . Let $S_{v,h}^{act}(\mathbb{s})$ denote the set of active sensors at t_h by applying algorithm \mathbb{s} and point v can be covered by these active sensors. Consider a space-time point (v, h) , $v \in R$ and $t_h \in T_c$. Let $p_{v,h}^{\mathbb{s}}$ denote the cooperative detection probability of all the sensors in $S_{v,h}^{act}(\mathbb{s})$ to space-time point (v, h) . The value of $p_{v,h}^{\mathbb{s}}$ can be calculated by applying Exp. (5).

$$p_{v,h}^{\mathbb{s}} = 1 - \prod_{s_i \in S_{v,h}^{act}(\mathbb{s})} (1 - p(s_i, v)) \quad (5)$$

Let \mathbb{C} denote the set of all the potential crossing paths, where $c_j \in \mathbb{C}$ represents the j -th crossing path in \mathbb{C} . Let $q_h(\mathbb{s}, j)$ denote the surveillance quality of $DB_{\mathbb{s}}$ to c_j at t_h . It is noted that an intruder traversing the monitored region along the crossing path c_j is considered effectively detected if the intruder is detected at least once at any point on c_j . Therefore, $q_h(\mathbb{s}, j)$ can be represented by the maximal cooperative detection probability of any point on c_j at t_h . That is

$$q_h(\mathbb{s}, j) = \max_{v \in c_j} p_{v,h}^{\mathbb{s}}$$

To guarantee the surveillance quality of $DB_{\mathbb{s}}$ can be maximized at all time slots, the surveillance quality of $DB_{\mathbb{s}}$ to c_j , denoted by $q(\mathbb{s}, j)$, is the weakest surveillance quality at any time slot in one cycle. That is

$$q(\mathbb{s}, j) = \min_{t_h \in T_c} q_h(\mathbb{s}, j)$$

Let $q(\mathbb{s})$ denote the surveillance quality of $DB_{\mathbb{s}}$. $q(\mathbb{s})$ can be represented by the weakest surveillance quality of $DB_{\mathbb{s}}$ to all potential crossing paths in \mathbb{C} . That is

$$q(\mathbb{s}) = \min_{c_j \in \mathbb{C}} q(\mathbb{s}, j)$$

However, both the number of points on c_j and the number of potential crossing paths are infinite. This paper applies the grid-based approach to cope with this problem. The detail of this approach is presented in section 4.

The proposed BCRAS algorithm aims to maximize the surveillance quality of the constructed barrier while balancing energy consumption and acquisition to perpetuate the network lifetime of deployed WRSNs. The object function of this paper is given in Exp. (6).

Objective Function

$$\max(q(\mathbb{s})) \quad (6)$$

Exp. (6) should be achieved under some constraints which are presented in the following.

Let Boolean variables $M_{j,h}^{sen-cha}$, $M_{j,h}^{sen-only}$, $M_{j,h}^{cha-only}$ and $M_{j,h}^{slep}$ represent whether sensor s_j stays in sensing & charging, sensing-only, charging-only, or sleeping states at t_h , respectively. That is

$$M_{j,h}^{sen-cha} = \begin{cases} 1, & \text{if } s_j \text{ stay in sensing \& charging state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M_{j,h}^{sen-only} = \begin{cases} 1, & \text{if } s_j \text{ stay in sensing - only state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M_{j,h}^{cha-only} = \begin{cases} 1, & \text{if } s_j \text{ stay in charging - only state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M_{j,h}^{slep} = \begin{cases} 1, & \text{if } s_j \text{ stay in sleeping - only state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

Exp. (7) gives the constraint that each sensor can only stay in one of these four states at any time slot.

(1) State Constraint

$$M_{j,h}^{sen-cha} + M_{j,h}^{sen-only} + M_{j,h}^{cha-only} + M_{j,h}^{slep} = 1 \quad (7)$$

The second constraint guarantees that each scheduled sensor should perform sensing operation for at least one time slot such that all sensors can participate in surveillance tasks. Exp. (8) reflects this requirement.

(2) Cycle-Working Constraint

$$\sum_{h=1}^{\frac{T_c}{\tau}} M_{j,h}^{sen-cha} + M_{j,h}^{sen-only} \geq 1 \quad \forall s_j \quad (8)$$

where τ is the length of one time slot.

Let $p^{PV}(t)$ denote the PV power at time t . The total energy that can be acquired from solar power in one day is $\int_{t_s}^{t_e} p^{PV}(t)dt$, where t_s and t_e are the starting and ending times of daytime, respectively. Conversely, the consumed energy of the sensor in one day is

$$\sum_{j=1}^T \frac{T}{T_c} (p_{i,k}^{sen} \cdot t_{i,k}^{sen}) \quad (9)$$

where T is the length of one day. To guarantee the perpetual lifetime of each sensor, the acquired energy and the consumed energy should be balanced each day. Exp. (10) reflects this constraint.

(3) Balancing the Energy Consumption and Acquisition Constraint

$$\int_{t_s}^{t_e} p^{PV}(t)dt = \sum_{j=1}^T \frac{T}{T_c} (p_{i,k}^{sen} \cdot t_{i,k}^{sen}) \quad (10)$$

The next section presents the proposed *BCRAS* algorithm which aims to achieve the objective function shown in Exp. (6) while satisfying constraints given in Exps. (7)-(10).

4 The Proposed *BCRAS* Algorithm

This paper proposes a novel barrier coverage algorithm, called *BCRAS*. The *BCRAS* algorithm is designed to efficiently schedule each sensor, enabling them to conduct sensing, charging, and sleeping operations in different time slots based on their sensing radius and available solar energy. This coordinated approach ensures effective monitoring of the given region, enhancing surveillance quality and perpetuating the network lifetime.

The *BCRAS* algorithm mainly consists of three phases: *Space-Time Partitioning Phase*, *Detection Probability Calculation Phase*, and *Sensor Scheduling Phase*. The first phase initially partitions the monitoring region R into several equal-sized grids. Additionally, to account for the variability of solar power and its susceptibility to weather conditions, the CNN-LSTM model is adopted to predict the PV power of the next day, aiming to estimate the available solar energy accurately. Subsequently, the time axis is divided into equal-length cycles. Each cycle consists of a fixed number of time slots. The length of each cycle and time slot is determined by the predicted available solar energy of the next day and the power consumption rate of sensors, aiming to balance the energy consumption with harvesting. Utilizing these identical time slots and grids, the *BCRAS* constructs a data structure that contains two-dimensional space-time points, making it easy to evaluate the detection probability of each sensor to each grid at every time slot. Then the *Detection Probability Calculation Phase* mainly calculates the detection probability of each sensor to each grid, representing the surveillance contribution of each sensor.

Finally, in the *Sensor Scheduling Phase*, *BCRAS* initially identifies the bottleneck space-time point, defined as the space-time point with the weakest cooperative detection probability. Subsequently, each sensor aims to independently construct the best task schedule in a distributed manner. The sensor with the maximum surveillance contribution to the bottleneck space-time point is prioritized for activation, such that the surveillance quality can be maximized.

4.1 Space-time Partitioning Phase

The main concept of this algorithm is to identify the bottleneck space-time points along all potential crossing paths at all times, and then construct the task schedule for each sensor, aiming to maximize the cooperative detection probability of all active sensors to bottlenecked space-time points while balancing energy consumption and acquisition to perpetuate network lifetime. However, the number of potential crossing paths is infinite. This implies that the number of possible space-time points is uncountable. To simplify the investigated issue and reduce computation

complexity, the grid-based approach is adopted. In this phase, the monitoring region R is first partitioned into $c_1 \times c_2$ equal-sized grids. The detection probability of any point within the same grid by each sensor is considered to be identical. Each grid will be labeled with two-dimensional coordinates. The most top-left grid in R is labeled with (1,1). The x -coordinate and y -coordinate are increased by one if the location of a grid shifts one position toward the right and down directions. Let $g_{(x,y)}$ denote the grid with coordinates (x, y) . Figure 4 shows an example of a monitoring region R , which is partitioned into 16×10 equal-sized grids. The sensors s_1, s_2 and s_3 are three active sensors for participating in a defense barrier.

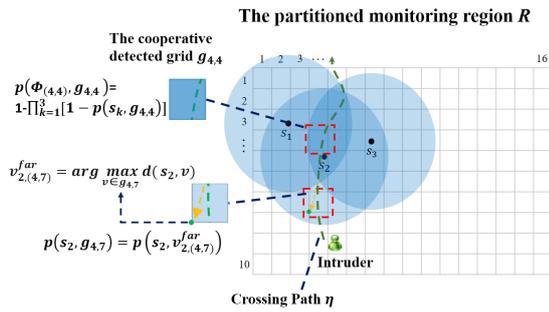


Figure 4. The partitioned monitoring region R based on the grid-based method

It is noticed that the number of time points in the time axis is also infinite. Based on the same considerations for space partitioning, the time axis will be partitioned into several equal-length cycles and each cycle consists of several equal-length time slots. The sensor will be scheduled to stay in different states at different time slots in each cycle. The length of each cycle and time slot depends on the available solar energy of the next day and the sensor's power consumption rate, aiming to balance the energy acquired and consumed by each sensor throughout the day. The available energy for the next day can be derived from the PV power for that day. However, the PV power of the next day is unknown, it needs to be predicted in advance.

The operations of time partitioning mainly consist of two steps. The first step is to employ the CNN-LSTM model to predict the PV power for the next day, such that the available solar power of the next day can be evaluated. The second step is to calculate the length of each time slot and cycle for both daytime and nighttime, based on the predicted available solar energy for the next day and the discharging rate of each sensor, aiming to balance the energy acquired and consumed.

Step 1: Predicting the PV power of the next day

The total amount of solar energy in a certain period is the integral of the PV power function. Therefore, the future PV power function should be forecasted in advance. It is noticed that PV power data is sequentially generated over time, characterized by high dimensionality and temporal dependency. High dimensionality means each time point is a dimension, while temporal dependency indicates that identical values may lead to different outcomes.

The irradiance, the primary factor affecting PV power, is influenced by weather conditions such as temperature, humidity, and cloud cover, which are spatially correlated. Convolutional Neural Networks (CNNs) are particularly effective in modeling these spatial correlation data. Moreover, the sequential data for these factors shows a strong temporal correlation [28], which Long Short-Term Memory (LSTM) networks handle effectively. Combining CNN and LSTM models leverages their strengths, leading to more accurate PV power predictions than using a single model [30]. This step applies a CNN-LSTM hybrid model for PV power prediction. Compared with other typical time series forecasting methods such as ARIMA, SVR, and standalone LSTM, the CNN-LSTM hybrid model offers superior performance in handling PV power data. ARIMA assumes linearity and struggles with the non-stationary nature of weather-influenced solar power series. SVR captures non-linear patterns but lacks temporal memory. Pure LSTM models handle time dependencies well but may overlook spatial correlations among meteorological inputs. By combining CNN's ability to extract spatial features with LSTM's temporal modeling strength, the CNN-LSTM architecture provides improved robustness and accuracy in dynamic and weather-driven energy forecasting scenarios.

To predict the next day's PV power accurately, while conserving computational resources, the CNN-LSTM model is pre-trained based on an existing mature dataset and operated at the base station, the base station will obtain the relevant meteorological data for the next day through the network, and then the prediction results will be transmitted to each sensor using a multi-hop approach.

Figure 5 shows the structure of the applied CNN-LSTM network, comprising CNN and LSTM components. The CNN extracts spatial features from input data like historical PV power and meteorological variables (irradiance, temperature, humidity, cloud cover) using convolutional and pooling layers. This is because PV power data alone reflects the output result of past conditions but does not explicitly capture the environmental drivers that influence solar energy generation. By incorporating these meteorological features, the model can better learn the causal relationships and improve the accuracy and robustness of PV power prediction under dynamic weather conditions. These features are then flattened and fed into the LSTM, which captures temporal dependencies with memory cells and gates (input, forget, output). The output of LSTM is transformed by fully connected layers into the final PV power prediction, enhancing accuracy.

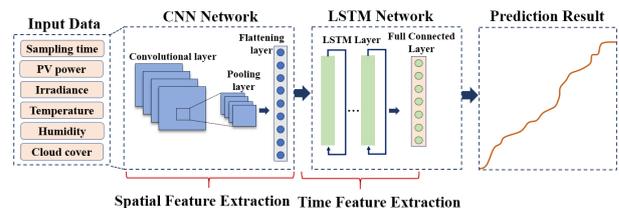


Figure 5. Structure of the proposed CNN-LSTM network

The following provides the process for predicting photovoltaic power using a CNN-LSTM hybrid model. Let $X_t = (x_{t,1}, x_{t,2}, \dots, x_{t,6})$ denote the input data which includes 6 features, where $x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}, x_{t,5}, x_{t,6}$ represent the sampling time, PV power, irradiance, temperature, humidity, and cloud cover, respectively. The following two expressions give the output of the first convolutional layer and the l -th convolutional layer, that is,

$$y_{ij}^1 = \sigma \left(b_j^1 + \sum_{m=1}^M W_{m,j}^1 x_{i+m-1,j}^0 \right)$$

$$y_{ij}^l = \sigma \left(b_j^l + \sum_{m=1}^M W_{m,j}^l x_{i+m-1,j}^0 \right)$$

where y_{ij}^l is calculated by the output vector x_{ij}^l of the previous layer. b_j^l represent the bias for the j -th feature map, w is the weight of the kernel, m is the index value of the filter, and σ is the activation function Relu.

Then the pooling layer reduces CNN parameters using maximum pooling. The maximum pooling operation is:

$$p_{i,j}^l = \max_{r \in R} y_{i \times T + r, j}^{l-1}$$

where R is the pooling size and satisfies $R < y$. T is the step size.

The output of the CNN is the input of LSTM. The LSTM stores time information of the main characteristics extracted from input data through the CNN network.

The LSTM consists of the input, forget, and output gates. Information is transferred by the cell over random time intervals. The gates trace the flow of the input and output data from the cell. The nodal outputs of an LSTM network are computed as follows:

$$i_t = \sigma \left(W_{p_i} p_t + W_{h_i} h_{t-1} + W_{c_i} \circ c_{t-1} + b_i \right)$$

$$f_t = \sigma \left(W_{p_f} p_t + W_{h_f} h_{t-1} + W_{c_f} \circ c_{t-1} + b_f \right)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma \left(W_{p_c} p_t + W_{h_c} h_{t-1} + b_c \right)$$

$$o_t = \sigma \left(W_{p_o} p_t + W_{h_o} h_{t-1} + W_{c_o} \circ c_{t-1} + b_o \right)$$

$$h_t = o_t \circ \sigma \left(c_t \right)$$

Where W represents the weight matrices of each gate. c and h represent the cell state and the hidden state is determined through the input, forget gate, and output gates. σ is the activation function. b is the bias vector. p_t contains the key features of photovoltaic power data, serving as the output of the pooling layer at time $t-1$ and as the input to the LSTM memory cell.

The final layer of the LSTM network is a fully connected layer that provides the prediction over a specific period. The LSTM output is flattened into a feature vector $h^l = [h_1, h_2, \dots, h_l]$, where l is the number of LSTM units. For PV power prediction at time t , the result y_t , output by

the fully connected layer, is calculated as follows:

$$x_{t+1} = \sum W_{ji}^{l-1} \left(\sigma \left(h_i^{l-1} \right) + b_i^{l-1} \right)$$

Step 2: Calculating the length of one time slot and Cycle

After obtaining the predicted PV power data of the next day, the corresponding PV Power function of the next day can be fitted according to some mature methods. Let $\mathcal{P}_i^{PV}(t)$ denote the fitted PV power function of the next day, which is the j -th day of this year. It is noticed that each sensor can only be charged during the daytime, therefore, parts of energy should be reserved to keep each sensor working properly such that the surveillance quality can be guaranteed in the night. Let E_{day}^{sen} and E_{night}^{sen} denote the energy allocated to each sensor for consuming in the daytime and nighttime, respectively. Let E_{cha} denote the amount of energy that is acquired in one day, that is

$$E_{cha} = \int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt$$

where t_s and t_e are the start time and end time of daytime.

Therefore, we have

$$E_{day}^{sen} + E_{night}^{sen} = \int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt$$

Recall that in the Charging and Discharging Model of section 3, this paper assumes that the energy allocated for each cycle is only sufficient to sustain the sensor being activated at the maximum sensing radius r_m for one time slot per cycle, which is the length of one time slot, we have

$$\tau = \frac{E_{day}^{sen}}{\mathcal{P}_{i,m}^{sen}} \tag{12}$$

where $\mathcal{P}_{i,m}^{sen}$ is the power consumption rate of the sensor s_i with maximum sensing radius r_m .

As shown in Figur 3, assume that each cycle consists of $\alpha + \beta$ time slots. During the daytime, α time slots are allocated for charging the energy consumed by the sensors throughout the daytime. It is noticed that each sensor can be continuously charged during the daytime. Therefore, the acquired and consumed energy can be balanced if the energy consumed in sensing & charging states can be equal to the energy charged in sensing & charging and charging-only states. This relationship is described by the Exp. (13).

$$\frac{\int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt}{t_e - t_s} \cdot \alpha \cdot \tau = \mathcal{P}_{i,m}^{sen} \cdot \tau \tag{13}$$

where $\frac{\int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt}{t_e - t_s}$ represents the average charging rate of PV power during the daytime. According to Exp. (13), the value of α can be obtained, that is

$$\alpha = \frac{\mathcal{P}_{i,m}^{sen}(t_e - t_s)}{\int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt} \quad (14)$$

The number of cycles in the daytime, denoted by N_c^{day} , is calculated as follows.

$$N_c^{day} = \frac{t_e - t_s}{(\alpha + \beta) \cdot \tau}$$

Therefore, the total consumed energy during the daytime is

$$E_{day}^{sen} = N_c^{day} \cdot \mathcal{P}_{i,m}^{sen} \cdot \tau$$

On the other side, in the daytime, β time slots within each cycle are allocated for charging energy consumed by sensors during nighttime. These time slots are illustrated by blue rectangles in Figure 3. During the nighttime, the cycle length is equal to that during the daytime. Consequently, the number of cycles in the nighttime, denoted by N_c^{Night} , is calculated as follows.

$$N_c^{night} = \frac{T - t_e + t_s}{(\alpha + \beta) \cdot \tau}$$

where T is the length of one day. Consequently, the total consumed energy during the nighttime is

$$E_{night}^{sen} = N_c^{night} \cdot \mathcal{P}_{i,m}^{sen} \cdot \tau$$

To guarantee the balance of E_{day}^{sen} and E_{night}^{sen} , we have

$$(N_c^{day} + N_c^{night}) \cdot \tau \cdot \mathcal{P}_{i,m}^{sen} = \int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt \quad (15)$$

Based on Exps. (14) and (15), the value of β is

$$\beta = \frac{\mathcal{P}_{i,m}^{sen} [T - t_e + t_s]}{\int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt}$$

Therefore, the length of one cycle is

$$T_c = \frac{T \cdot \tau \cdot \mathcal{P}_{i,m}^{sen}}{\int_{t_s}^{t_e} \mathcal{P}_i^{PV}(t) dt}$$

Till now, the length of one time slot τ and one cycle T_c are determined. In the next phase, the detection probability of each sensor to each grid in R is further calculated.

4.2 Detection Probability Calculation Phase

This phase aims to calculate the detection probability of each sensor to each grid in R , which is essential for assessing surveillance quality. Assume that sensor s_i can cover grid $\mathcal{G}_{(x,y)}$. The detection probability depends on the

situation whether or not the $\mathcal{G}_{(x,y)}$ is fully covered by s_i . In case a grid $\mathcal{G}_{(x,y)}$ is not fully covered by the sensor s_i , any intruder that located at the uncovered part of $\mathcal{G}_{(x,y)}$ will remain undetected. To guarantee the surveillance quality, the detection probability of the sensor s_i to $\mathcal{G}_{(x,y)}$ is 0 in this case.

On the contrary, if the sensor s_i can fully cover the grid $\mathcal{G}_{(x,y)}$, the detection probability of s_i to $\mathcal{G}_{(x,y)}$ depends on the distance between s_i and $\mathcal{G}_{(x,y)}$. Let $p(s_i, \mathcal{G}_{(x,y)})$ denote the detection probability of s_i to $\mathcal{G}_{(x,y)}$. Let $v_{i,(x,y)}^{far}$ denote the farthest point within $\mathcal{G}_{(x,y)}$ from s_i . That is

$$v_{i,(x,y)}^{far} = \arg \max_{v \in \mathcal{G}_{(x,y)}} d(v, s_i)$$

To guarantee the surveillance quality, the point $v_{i,(x,y)}^{far}$ is selected as the representative point of the grid $\mathcal{G}_{(x,y)}$. Therefore, the detection probability of s_i to $\mathcal{G}_{(x,y)}$ can be calculated by applying Exp. (16).

$$p(s_i, \mathcal{G}_{(x,y)}) = p(s_i, v_{i,(x,y)}^{far}) \quad (16)$$

The following further discusses the case when two or more active sensors simultaneously cover the grid $\mathcal{G}_{(x,y)}$. Let $\Phi_{(x,y)} = \{s_1^{sen}, s_2^{sen}, \dots, s_m^{sen}\}$ denote the set of active sensors that fully cover the grid $\mathcal{G}_{(x,y)}$. Let $p^{co}(\Phi_{(x,y)}, \mathcal{G}_{(x,y)})$ denote the cooperative detection probability of all sensors in $\Phi_{(x,y)}$ to $\mathcal{G}_{(x,y)}$. The value of $p(\Phi_{(x,y)}, \mathcal{G}_{(x,y)})$ can be calculated by applying Exp. (17).

$$p^{co}(\Phi_{(x,y)}, \mathcal{G}_{(x,y)}) = 1 - \prod_{k=1}^m [1 - p(s_k^{sen}, \mathcal{G}_{(x,y)})] \quad (17)$$

As shown in Figure 4, grid $\mathcal{G}_{(2,7)}$ is not fully covered by the sensor s_2 , we have $p(s_2, \mathcal{G}_{(2,7)}) = 0$. On the contrary, the grid $\mathcal{G}_{(4,7)}$ is fully covered by the sensor s_2 , the farthest point in $\mathcal{G}_{(4,7)}$ to s_2 is marked by the green point as shown in Figure 4. The detection probability of s_2 to $\mathcal{G}_{(4,7)}$ depends on the distance between s_2 and $v_{2,(4,7)}^{far}$. By applying the sensing model presented in Exp. (1), we have

$$p(s_2, \mathcal{G}_{(4,7)}) = p(s_2, v_{2,(4,7)}^{far}) = \begin{cases} 1 & \text{if } d(s_2, v_{2,(4,7)}^{far}) \leq r_g \\ e^{-\lambda(d(s_2, v_{2,(4,7)}^{far}) - r_g)^\beta} & \text{if } r_g < d(s_2, v_{2,(4,7)}^{far}) \leq r_m \\ 0 & \text{if } d(s_2, v_{2,(4,7)}^{far}) > r_m \end{cases}$$

Furthermore, grid $\mathcal{G}_{(4,4)}$ is fully covered by sensors s_1, s_2 and s_3 . That is, we have $\Phi_{(4,4)} = \{s_1, s_2, s_3\}$. The cooperative detection probability of s_1, s_2 and s_3 to $\mathcal{G}_{(4,4)}$ is

$$p^{co}(\Phi_{(4,4)}, \mathcal{G}_{(4,4)}) = 1 - \prod_{k=1}^3 [1 - p(s_k, \mathcal{G}_{(4,4)})]$$

In this phase, the detection probability of active sensors to each grid can be obtained. This detection probability will be used to identify the bottleneck space-time point and construct the task schedule for each sensor in the next phase.

4.3 Sensor Scheduling Phase

In this phase, each sensor aims to establish the best task schedule to maximize the surveillance quality of the constructed barrier while balancing energy consumption and acquisition to perpetuate the network lifetime. The principle of the proposed algorithm is to iteratively select the grid with the highest potential to obtain the maximum cooperative detection probability in each column of the partitioned region R as the target grid from B_{left} to B_{right} . Then the *BCRAS* algorithm aims to design the best task schedule for each sensor based on the current surveillance quality. Given that the surveillance schedule operates in a distributed manner, each sensor only needs to locally consider the monitoring region that falls in its local field of view (*FOV*).

Let FOV_i denote the local field of view of s_i , which can be expressed by the following expression

$$FOV_i = \{g_v | d(s_i, g_v) \leq r_m\}$$

Let $\mathcal{G}^{tar} = \{g_1^{tar}, g_2^{tar}, \dots, g_k^{tar}\}$ denote the set of target grids. Consider a space-time point (g_v^{tar}, t_h) , where $g_v^{tar} \in \mathcal{G}^{tar}$ and $t_h \in T_c$. Let R_i^{st} denote the local monitoring space-time region of s_i , which is spanned in two dimensions. In the space dimension, the R_i^{st} considers only the local field of view of the sensor s_i . In addition, in the time dimension, R_i^{st} only considers the time slot in a cycle T_c . The following expression depicts the two-dimensional space-time region of R_i^{st} .

$$R_i^{st} = \{(g_{i_v}^{tar}, t_h) | g_{i_v}^{tar} \in FOV_i, t_h \in T_c\}$$

A sensor s_j is said to be the neighboring sensor of s_i if the Euclidean distance between s_i and s_j is less than the communication radius of each sensor. Recall that the communication radius is assumed twice the maximum sensing radius. Let $\mathbb{N}(s_i)$ denote the set of the neighboring sensors of s_i , that is

$$\mathbb{N}(s_i) = \{s_j | d(s_i, s_j) \leq 2r_m\}$$

Let $p_{v,h}$ denote the cooperative detection probability of all the active sensors to the space-time point (g_v^{tar}, t_h) , which reflects the cooperative detection probability contributed by the set of active sensors that can fully cover the grid g_v^{tar} at time slot t_h . By applying the product theorem in probability theory, we have

$$p_{v,h} = 1 - \prod_{s_j \in S_{v,h}^{sch}} [1 - p_{v,h}^{s_j}] \quad (18)$$

where $S_{v,h}^{sch}$ is the set of sensors that can fully cover g_v^{tar} and

are scheduled to be active at the time slot t_h , and $p_{v,h}^{s_j}$ is the detection probability of the sensor s_j to space-time point (g_v^{tar}, t_h) .

Let $o_{i,loc}^{weak}$ denote the local bottleneck space-time point in R_i^{st} , which is the space-time point with the weakest cooperative detection probability in R_i^{st} . That is

$$o_{i,loc}^{weak} = \arg \min_{(g_{i_v}^{tar}, t_h) \in R_i^{st}} (p_{i_v,h}) \quad (19)$$

The task schedule is cycle-based. This indicates that all cycles have the same schedule. The Sensor Scheduling Phase mainly consists of four steps. The first step selects the set of target grids from B_{left} to B_{right} . The second step evaluates the surveillance quality of each space-time point and finds the local bottleneck space-time point $o_{i,loc}^{weak}$ in R_i^{st} . Then in the third step, each sensor s_i aims to construct the best task schedule that can maximize the surveillance quality of $o_{i,loc}^{weak}$ in R_i^{st} . In the last step, each sensor cooperates with its neighbors to achieve the maximal surveillance quality.

Step 1: Selecting the set of the target grid

This step initially selects the first target grid g_1^{tar} which is located in the leftmost column of R . Then the proposed *BCRAS* selects the next target grid g_{n+1}^{tar} in each column such that the selected grid is connected to the target g_n^{tar} . This step will be executed round by round until the right boundary B_{right} is reached.

For any grid $g_{(x,y)}$ in R , recall that the $\Phi_{(x,y)}$ denotes the set of sensors that can fully cover $g_{(x,y)}$. It is noticed that the cooperative detection probability of the grid $g_{(x,y)}$ highly depends on two factors: the number of sensors belongs to $\Phi_{(x,y)}$ and the distances between each sensor in $\Phi_{(x,y)}$ and grid $g_{(x,y)}$. Although the scheduled sensor might utilize different sensing radiuses in different time slots, this step only considers the maximal sensing radius. It is because this step aims to find a set of grids that are most likely to obtain the maximum cooperative detection probability. The algorithm first calculates the cooperative detection probabilities of $g_{(1,j)}$ contributed by the sensors belonging to $\Phi_{(1,j)}$. The grid with the maximum cooperative detection probability will be selected as the first target grid g_1^{tar} . That is,

$$g_1^{tar} = \arg \max_{\forall g_{(1,j)}} [p^{co}(\Phi_{(1,j)}, g_{(1,j)})] \quad (20)$$

Figure 6 gives an example of selecting the first target grid. As shown in Figure 6, the monitoring region R is partitioned into 5×4 equal-sized grids. A set of sensors $S = \{s_1, s_2, \dots, s_{13}\}$ are randomly deployed in R . Assume that all the sensors in S are activated with the maximum sensing radius. The proposed *BCRAS* initially calculates the cooperative detection probability of each grid which is located in the leftmost column. The values of $p^{co}(\Phi_{(1,1)}, g_{(1,1)})$, $p^{co}(\Phi_{(1,2)}, g_{(1,2)})$, $p^{co}(\Phi_{(1,3)}, g_{(1,3)})$, and $p^{co}(\Phi_{(1,4)}, g_{(1,4)})$ are labeled in the corresponding grid and are marked with blue color. The value of $p^{co}(\Phi_{(1,2)}, g_{(1,2)})$ is maximal, as compared with the other values. Therefore, the grid $g_{(1,2)}$ is selected as the g_1^{tar} which is marked with green color.

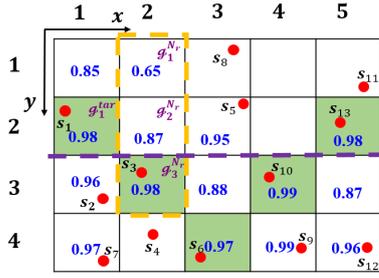


Figure 6. An example of selecting the target grid

As soon as the scheduling operation for the first target grid has been accomplished, the next work is to select the next target grid. To guarantee that the sensing radii of the active sensors can form a continuous barrier without any coverage hole, the next target grid should be selected from the right neighboring grids of the current target grid. The following defines the right neighboring grids of $\mathcal{G}_{(l,k)}$. The grid $\mathcal{G}_{(m,n)}$ is said to be the right neighboring grid of $\mathcal{G}_{(l,k)}$ if it satisfies Exp. (21).

$$m - l = 1 \ \&\& \ |n - k| \leq 1 \quad (21)$$

Under this definition, it is obvious that each grid has three right neighboring grids. Let $N_r(\mathcal{G}_{(x,y)}) = \{\mathcal{G}_1^{Nr}, \mathcal{G}_2^{Nr}, \mathcal{G}_3^{Nr}\}$ denote the set of right neighboring grids of $\mathcal{G}_{(x,y)}$. Assume the current target grid is \mathcal{G}_i^{tar} . Initially, all grids $\mathcal{G}_k^{Nr} \in N_r(\mathcal{G}_i^{tar})$ will be the candidates of the next target grid. Then the BCRAS calculates the potential cooperative detection probability of each \mathcal{G}_k^{Nr} . The grid $\mathcal{G}_k^{Nr} \in N_r(\mathcal{G}_i^{tar})$ with the maximum potential cooperative detection probability will be selected as the \mathcal{G}_{i+1}^{tar} . That is,

$$\mathcal{G}_{i+1}^{tar} = \arg \max_{\mathcal{G}_{(x,y)} \in N_r(\mathcal{G}_i^{tar})} [p^{co}(\Phi_{(x,y)}, \mathcal{G}_{(x,y)})] \quad (22)$$

Figure 6 additionally gives an example to illustrate the operations for selecting the next target grid. As shown in Figure 6, the $\mathcal{G}_{(1,2)}$ has been selected as the \mathcal{G}_1^{tar} . According to Exp. (21), we have $N_r(\mathcal{G}_1^{tar}) = \{\mathcal{G}_1^{Nr}, \mathcal{G}_2^{Nr}, \mathcal{G}_3^{Nr}\}$, which is located in the yellow dotted box as shown in Figure 6. Then the BCRAS calculates the cooperative detection probability of each grid of $N_r(\mathcal{G}_1^{tar})$. The value of $p^{co}(\Phi_{(2,1)}, \mathcal{G}_{(2,1)})$, $p^{co}(\Phi_{(2,2)}, \mathcal{G}_{(2,2)})$, $p^{co}(\Phi_{(2,3)}, \mathcal{G}_{(2,3)})$ are labeled in the corresponding grids and are marked with blue color. According to Exp. (22), the grid $\mathcal{G}_{(2,3)}$ is selected as the second target grid.

In case there exists more than one grid that satisfies Exp. (20) when selecting the first target grid or satisfies Exp. (22) when selecting the next target grid, the grid closest to the central top-down line of the region will be finally selected as the target grid. This is because the target grid closer to the central line is more likely to be fully covered by more sensors. This policy can select the grid which has a higher cooperative detection probability. As shown in Figure 6, the purple line is the central top-down line of R . Assume that $\mathcal{G}_{(3,4)}$ is the current selected target grid. Both $\mathcal{G}_{(4,3)}$ and $\mathcal{G}_{(4,4)}$ are the right neighboring grids of $\mathcal{G}_{(3,4)}$. The coopera-

tive detection probabilities of $\mathcal{G}_{(4,3)}$ and $\mathcal{G}_{(4,4)}$ are both 0.99. Then the $\mathcal{G}_{(4,3)}$ is selected as the next target grid because $\mathcal{G}_{(4,3)}$ is closer to the central line than $\mathcal{G}_{(4,4)}$.

Based on the operations mentioned above, the set of continuous target grids from B_{left} to B_{right} can be obtained accordingly. Each target grid and time slot consists of a set of space-time points that are to be monitored by the constructed barrier. Then the next step is to identify the ‘‘bottleneck space-time point’’ $o_{i,loc}^{weak}$ in R_i^{st} .

Step 2: Identification of the ‘‘bottleneck space-time point’’

Recall S is the set of all the deployed sensors in the monitoring region. Each sensor $s_i \in S$ will establish a matrix to maintain the local surveillance quality of current active sensors to each space-time point in R_i^{st} . Assume that there are ℓ_i target grids locate in FOV_i . Let \mathcal{M}_i^{QoS} denote the surveillance quality matrix of R_i^{st} which is expressed as follows:

$$\mathcal{M}_i^{QoS} = \begin{pmatrix} P_{i,1} & \cdots & P_{i,n} \\ \vdots & \ddots & \vdots \\ P_{i_{1+\ell_i},1} & \cdots & P_{i_{1+\ell_i},n} \end{pmatrix} \quad (23)$$

Initially, each sensor broadcasts its current task schedule. Meanwhile, sensor s_i obtains the current task schedules from its neighboring sensors. Let $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$ denote the received schedules matrix which is maintained by s_i . The $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$ is expressed in the following form:

$$\mathcal{M}_{\mathbb{N}(s_i)}^{sch} = \begin{pmatrix} S_{i,1}^{sch} & \cdots & S_{i,n}^{sch} \\ \vdots & \ddots & \vdots \\ S_{i_{1+\ell_i},1}^{sch} & \cdots & S_{i_{1+\ell_i},n}^{sch} \end{pmatrix} \quad (24)$$

According to the matrix $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$, each element in \mathcal{M}_i^{QoS} can be evaluated by applying Exp. (18). Then based on matrices \mathcal{M}_i^{QoS} , the local bottleneck space-time point $o_{i,loc}^{weak}$ in R_i^{st} can be obtained by applying Exp. (19). It is noted that the number of bottleneck points might be more than one. Therefore, let $o_{i,loc}^{weak}$ denote the set of local bottleneck space-time points in R_i^{st} . Then in the next step, the best surveillance schedule for each sensor s_i will be constructed in a distributed manner, aiming to improve the surveillance quality of each R_i^{st} .

Figure 7 first depicts an example to illustrate the identification of the ‘‘bottleneck space-time point’’. As shown in Figure 7, for sensor s_3 , the selectable sensing radiuses of s_3 is $\mathbb{R} = \{r_1, r_2, r_3\}$. Therefore, the local field of view of s_3 is

$$FOV_3 = \{\mathcal{G}_v | d(s_3, \mathcal{G}_v) \leq r_3\}$$

where the grids in FOV_3 are located in the area surrounded by the red line in Figure 7. Therefore, the local monitoring space-time region of s_3 is

$$R_3^{st} = \{(\mathcal{G}_{3v}^{tar}, t_h) \mid \mathcal{G}_{3v}^{tar} \in FOV_3, t_h \in T_c\}$$

where T_c is assumed to be $T_c = \{t_1, t_2, t_3\}$. The neighboring sensors of s_3 is $\mathbb{N}(s_3) = \{s_1, s_2, \dots, s_{13}\}$, which are the red points in Figure 7.

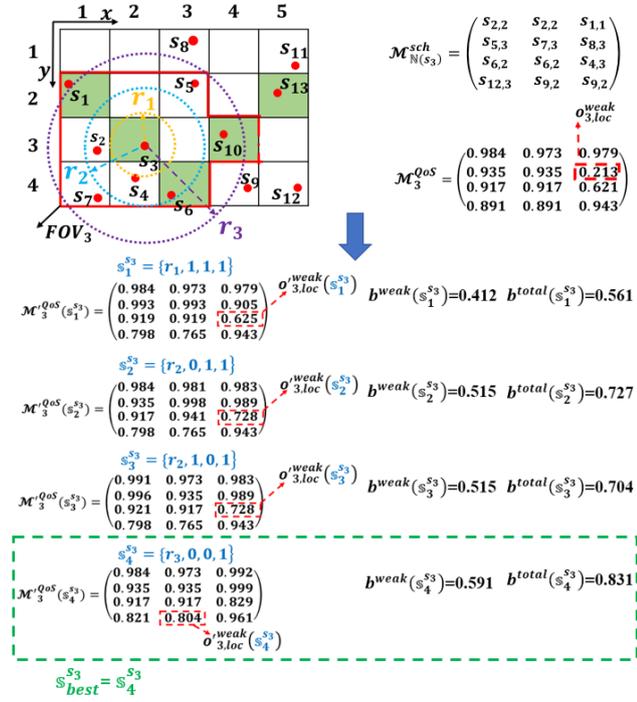


Figure 7. An example of the operations in step 2 and step 3

Assume that the current schedules matrix of s_3 is

$$\mathcal{M}_{\mathbb{N}(s_3)}^{sch} = \begin{pmatrix} s_{2,2} & s_{2,2} & s_{1,1} \\ s_{5,3} & s_{7,3} & s_{8,3} \\ s_{6,2} & s_{6,2} & s_{4,3} \\ s_{12,3} & s_{9,2} & s_{9,2} \end{pmatrix}$$

Based on $\mathcal{M}_{\mathbb{N}(s_3)}^{sch}$, the current surveillance quality matrix of R_3^{st} can be obtained, that is

$$\mathcal{M}_3^{QoS} = \begin{pmatrix} 0.984 & 0.973 & 0.979 \\ 0.935 & 0.935 & \mathbf{0.213} \\ 0.917 & 0.917 & 0.621 \\ 0.891 & 0.891 & 0.943 \end{pmatrix} \quad (25)$$

According to Exp. (19), the local bottleneck space-time points of R_3^{st} is $o_{3,loc}^{weak} = (\mathcal{G}_{2,2}^{tar}, t_3)$, which is marked in red ink in Exp. (25).

Step 3: Constructing the best schedule for each sensor.

This step aims to design the best task schedule for each sensor s_i for improving the cooperative detection probability of $o_{i,loc}^{weak}$, such that the local surveillance quality of R_i^{st} can be maximized. It is noticed that the range of

distance between each sensor and each target grid varies greatly. In this paper, the sensing radius that can cover any target grid is called the usable sensing radius. Therefore, the farther the sensor is from the target grid, the less usable sensing radius can be selected. The sensors that are farthest from the set of target grids may not cover any of them unless they adopt the maximum sensing radius. However, these target grids may not be the bottleneck space-time points if some other sensors are already scheduled to cover them. To give all the unscheduled sensors more opportunities to make contributions to the bottleneck space-time points, the Farthest Distance Scheduling First (FDSF) policy is proposed, the unscheduled sensor that exhibits the maximum distance from the target grids is prioritized for scheduling.

According to the FDSF policy, the sensors with the same number of usable sensing radii have the same priority. Therefore, the operations of this step will be executed round by round. The basic operations of constructing the task schedule in each round are similar, which is to decide the sensing radius of each sensor and the state that each sensor stays in at each time slot, aiming to maximize the surveillance quality while balancing energy consumption and acquisition.

Assume that any sensor s_i categorized in the k -th round to schedule has k usable sensing radius. It is noticed that any task schedule for s_i includes the decisions of sensing radius of s_i and the state of s_i at each time slot in a cycle, which can be represented by a vector. Let $s_l^{s_i}$ denote a task schedule for the sensor s_i , that is

$$s_l^{s_i} = (r_l, t_{l,1}^{s_i}, t_{l,2}^{s_i}, \dots, t_{l,m}^{s_i})$$

where $r_l \in \mathbb{R}$, and any $t_{l,h}^{s_i}$ in $s_l^{s_i}$ is the Boolean variable that represents the state of s_i at time slot t_h , that is,

$$t_{l,h}^{s_i} = \begin{cases} 1, & \text{if } s_i \text{ is activated at } t_v \text{ under task schedule } s_l^{s_i} \\ 0, & \text{otherwise} \end{cases}$$

Recall that the active time of each sensor should satisfy Exp. (4). That is, each legal task schedule $s_l^{s_i}$ must satisfy the following expression:

$$\sum_{h=1}^m t_{l,h}^{s_i} = m - l + 1$$

where m is the number of the selectable sensing radii of each sensor.

Let s^{s_i} denote the set of all possible legal task schedules for s_i . In case that s_i selects $s_l^{s_i} \in s^{s_i}$ as the task schedule, the local surveillance quality matrix \mathcal{M}_i^{QoS} should be updated accordingly. To obtain the best task schedule among all possible schedules in s^{s_i} , Exp. (26) initially depicts the updated local surveillance quality matrix $\mathcal{M}_i^{QoS}(s_l^{s_i})$.

$$\mathcal{M}_i^{QoS}(\mathfrak{s}_l^{s_i}) = \begin{pmatrix} p_{i,1}(\mathfrak{s}_l^{s_i}) & \cdots & p_{i,n}(\mathfrak{s}_l^{s_i}) \\ \vdots & \ddots & \vdots \\ p_{i+\kappa_i,1}(\mathfrak{s}_l^{s_i}) & \cdots & p_{i+\kappa_i,n}(\mathfrak{s}_l^{s_i}) \end{pmatrix} \quad (26)$$

$$= \begin{pmatrix} p_{i,1}(\mathfrak{s}_l^{s_i}) - p_{i,1} & \cdots & p_{i,n}(\mathfrak{s}_l^{s_i}) - p_{i,n} \\ \vdots & \ddots & \vdots \\ p_{i+\kappa_i,1}(\mathfrak{s}_l^{s_i}) - p_{i+\kappa_i,1} & \cdots & p_{i+\kappa_i,n}(\mathfrak{s}_l^{s_i}) - p_{i+\kappa_i,n} \end{pmatrix}$$

where each element $p_{i,v,h}(\mathfrak{s}_l^{s_i})$ in matrix $\mathcal{M}_i^{QoS}(\mathfrak{s}_l^{s_i})$ can be obtained by applying the following expression.

$$p_{i,v,h}(\mathfrak{s}_l^{s_i}) = 1 - (1 - p_{i,v,h})(1 - p_{i,v,h}^{s_i} \times t_{i,h}^{s_i})$$

where $p_{i,v,h}^{s_i}$ is the detection probability of the sensor s_i to space-time point $(\mathcal{G}_{i,v}^{tar}, t_h)$ when s_i select task schedule $\mathfrak{s}_l^{s_i}$.

Then based on the matrix $\mathcal{M}_i^{QoS}(\mathfrak{s}_l^{s_i})$, the new local space-time point in R_i^{st} if s_i adopt task schedule $\mathfrak{s}_l^{s_i}$ is calculated as follows:

$$o_{i,loc}^{weak}(\mathfrak{s}_l^{s_i}) = \arg \min_{(\mathcal{G}_{i,v}^{tar}, t_h) \in R_i^{st}} (p_{i,v,h}(\mathfrak{s}_l^{s_i})) \quad (27)$$

The primary purpose of this step is to maximize the surveillance quality of local bottleneck space-time point. Let $b^{weak}(\mathfrak{s}_l^{s_i})$ denote the main benefit of the sensor s_i selecting task schedule $\mathfrak{s}_l^{s_i}$, which is shown in the Exp. (28).

$$b^{weak}(\mathfrak{s}_l^{s_i}) = p_{i,v,h'}^{weak}(\mathfrak{s}_l^{s_i}) - p_{i,v,h'}^{weak} \quad (28)$$

where $p_{i,v,h'}^{weak}$ and $p_{i,v,h'}^{weak}(\mathfrak{s}_l^{s_i})$ are the cooperative detection probability of $o_{i,loc}^{weak}$ and $o_{i,loc}^{weak}(\mathfrak{s}_l^{s_i})$, respectively. $b^{weak}(\mathfrak{s}_l^{s_i})$ reflects the improved surveillance quality at bottleneck space-time point in R_i^{st} when s_i selects task schedule $\mathfrak{s}_l^{s_i}$.

Then Exp. (29) reflects the primary objective of step 3, which aims to maximize the improved surveillance quality of bottleneck space-time point.

$$\mathfrak{s}_{best}^{s_i} = \arg \max_{\mathfrak{s}_l^{s_i} \in \mathfrak{S}^{s_i}} (b^{weak}(\mathfrak{s}_l^{s_i})) \quad (29)$$

where $\mathfrak{s}_{best}^{s_i}$ is the best task schedule for s_i .

However, there may exist more than one task schedule that satisfies Exp. (29). Let $\mathfrak{S}_{best}^{s_i,cand}$ denote the set of the best task schedule candidates. In addition to the primary objective of eliminating the bottleneck space-time point, the secondary objective of step 3 aims to improve the cooperative detection probability of all space-time points in R_i^{st} . Let $b^{total}(\mathfrak{s}_l^{s_i})$ denote the total benefit of the sensor s_i selecting task schedule $\mathfrak{s}_l^{s_i}$, which is the amount of improved cooperative detection probability of all the space-time points in R_i^{st} . To calculate $b^{total}(\mathfrak{s}_l^{s_i})$, the algorithm first establishes a matrix $B(\mathfrak{s}_l^{s_i})$ to store the value of improved cooperative detection probability of all the space-time points in R_i^{st} , that is

$$B(\mathfrak{s}_l^{s_i}) = \mathcal{M}_i^{QoS}(\mathfrak{s}_l^{s_i}) - \mathcal{M}_i^{QoS}$$

Therefore, the value of $b^{total}(\mathfrak{s}_l^{s_i})$ can be obtained by applying the following expression,

$$b^{total}(\mathfrak{s}_l^{s_i}) = \sum_{u=1}^{\kappa_i} \sum_{h=1}^n [p_{i+u,h}(\mathfrak{s}_l^{s_i}) - p_{i+u,h}] \quad (30)$$

where any term $[p_{i+u,h}(\mathfrak{s}_l^{s_i}) - p_{i+u,h}]$ is the element in $B(\mathfrak{s}_l^{s_i})$.

Finally, the best task schedule of s_i is

$$\mathfrak{s}_{best}^{s_i} = \arg \max_{\mathfrak{s}_l^{s_i} \in \mathfrak{S}_{best}^{s_i,cand}} [b^{total}(\mathfrak{s}_l^{s_i})] \quad (31)$$

Up to now, the best task schedule $\mathfrak{s}_{best}^{s_i}$ of each sensor s_i for improving the surveillance quality of each R_i^{st} can be uniquely constructed in a distributed manner. However, the schedules of neighboring sensors might contradict each other. Besides, the broadcast packets containing the local decision of the task schedule might have collisions. To address these problems, the Random Back-off and Decision-Announcement Scheme will be presented in the next step.

To illustrate the operation in this step, the following further gives an example that continues the example shown in Figure 7. It is noticed that the s_8 is farthest from the target grids within FOV_3 , and it can only cover the target grid by selecting the maximum sensing radius r_3 . According to the *FDSF* policy, s_8 is categorized into the initial round of constructing the best schedule, which is already scheduled in this example. For sensor s_3 , it is positioned within the target grid \mathcal{G}_2 of FOV_3 , which is the closest to the target grid. Therefore, s_3 is categorized into the final round of task scheduling. The following continues to illustrate the operations of constructing the best schedule for s_3 . The operations of constructing the best task schedule for other sensors are identical to those of s_3 .

Recall the example in step 2, the local bottleneck space-time point in R_3^{st} is $o_{3,loc}^{weak} = (\mathcal{G}_2^{tar}, t_3)$. All the possible task schedules that can make contribution to $o_{3,loc}^{weak}$ are listed as follows.

$$\mathfrak{s}_1^{s_3} = \{r_1, 1, 1, 1\}$$

$$\mathfrak{s}_2^{s_3} = \{r_2, 0, 1, 1\}$$

$$\mathfrak{s}_3^{s_3} = \{r_2, 1, 0, 1\}$$

$$\mathfrak{s}_4^{s_3} = \{r_3, 0, 0, 1\}$$

For each possible task schedule, the local surveillance quality matrix \mathcal{M}_3^{QoS} will be updated accordingly, and the algorithm calculates each temporal surveillance quality

matrix $\mathcal{M}_3^{QoS}(\mathbb{S}_i^{s_3})$ for each possible task schedule $\mathbb{S}_i^{s_3}$, which are shown in Figure 7. Then the new local space-time point $o_{i,loc}^{weak}(\mathbb{S}_i^{s_3})$ if s_3 adopt task schedule $\mathbb{S}_i^{s_3}$ can be obtained by applying Exp. (27), which is marked by a red dotted box in Figure (9). For each task schedule $\mathbb{S}_i^{s_3}$, the main benefit $b^{weak}(\mathbb{S}_i^{s_3})$ and the total benefit $b^{total}(\mathbb{S}_i^{s_3})$ of s_3 selecting task schedule $\mathbb{S}_i^{s_3}$ can be calculated by applying Exps. (28) and (30), respectively.

Finally, the best task schedule $\mathbb{S}_{best}^{s_3}$ can be determined by applying Exps. (29) and (31), which is $\mathbb{S}_4^{s_3}$, and marked in green ink in Figure (9).

Step 4: The Random Back-off and Decision-Announcement Schemes

In this step, each sensor s_i needs to wait for a random time after it selects the best task schedule in each round of scheduling sensors. The major reason for applying the random back-off scheme is illustrated in the following. First, the task schedule $\mathbb{S}_{best}^{s_i}$ of s_i might contradict the task schedule $\mathbb{S}_{best}^{s_j}$ of its neighboring sensor s_j , where $s_j \in \mathbb{N}(s_i)$. The random back-off scheme [25] will be applied to obtain a better schedule. Another important reason for applying the random back-off scheme is to avoid collisions occurring among neighbors when broadcast packets contain the local decision of the task schedule.

The random waiting time of each sensor s_i should be determined based on the benefit of its own task schedule in terms of the surveillance quality. Let RB_i denote the random back-off time of s_i . Then the value of RB_i can be calculated by the following expression,

$$RB_i = [\omega_1 \cdot b^{bott}(\mathbb{S}_i^{s_i}) + \omega_2 \cdot b^{total}(\mathbb{S}_i^{s_i})]^{-1} \quad (32)$$

Where ω_1 is much larger than ω_2 , which reflects the primary purpose is to get higher surveillance quality of bottleneck space-time points, and the secondary purpose is to improve the cooperative detection probability of all space-time points.

After finishing the task schedule construction in each round, each sensor s_i should calculate its own RB_i and then countdown from its back-off time RB_i to 0. In case there is no schedule broadcasted from neighbors, the sensor s_i can broadcast its schedule. All sensors that are losers in the random back-off procedure will apply the task schedule broadcasted by the winner. According to the received schedule, all losers should update their local surveillance quality matrix \mathcal{M}_i^{QoS} and received schedules matrix $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$, then redesign its task schedule according to the operations presented in step 3.

Table 2 summarizes the process of the *BCRAS* algorithm. The computational complexity of the proposed algorithm is discussed in the following. The proposed *BCRAS* algorithm mainly consists of three phases. The *space-time partitioning phase* is summarized in steps 1 to 3 while the *detection probability calculation phase* is summarized in step 4. Finally, steps 5 to 8 summarize the *sensor scheduling phase*. In the first phase, the partition

of space and time takes constant time. Therefore, the complexity of steps 1 to 3 is $O(1)$. In the second phase, the complexities of Exps. (16) and (17) are $O(m*n)$ and $O(n)$, respectively. Therefore, the total complexity of the second phase is $O(m*n) + O(n)$. In the last phase, the complexities of Exps. (20) and (22) are $O(n)$ and $O(3)$, respectively. The complexities of Exps. (24) and (19) are $O(m*n*K)$ and $O(m*n)$, respectively, where K is the number of active sensors that can cover the space-time points in R_i^{st} . In step 7, the complexity of Exps. (29) and (31) is $O(m*n*|\mathbb{S}^{s_i}|)$, where $|\mathbb{S}^{s_i}|$ is the number of possible task schedules of s_i . In the last step, the operation of Exp. (32) takes a constant time $O(1)$. Consequently, the complexity of the third phase is $O(n) + O(m*n*K) + O(m*n*|\mathbb{S}^{s_i}|)$. In the real application, the number of grids is much larger than K and $|\mathbb{S}^{s_i}|$. Therefore, the complexities of all the phases on the sensor side can be simplified as $O(m*n)$. Therefore, y , the total complexity of the proposed *BCRAS* algorithm is simplified as $O(m*n)$.

Table 2. The *BCRAS* algorithm

Inputs:	
1.	A set of solar power sensors with adjustable sensing radius $S = \{s_1, s_2, \dots, s_n\}$.
2.	The historical meteorological data $\mathbf{X}^{MD} = \{\mathbf{x}_{1-n}, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$
3.	Some main features of the sensor: $E^{max}, E^{min}, R, \rho_{i,m}^{sen}$
Outputs: The scheduled sensors form a barrier coverage	
Step 1. /* Space Partitioning */	
	Partition the R into $c_1 \times c_2$ equal-sized grids;
Step 2. /* Predicting the PV power of the next day */	
	Predicting <i>PV</i> power of the next day by CNN-LSTM model;
Step 3. /* Calculating the Length of One Time Slot and Cycle */	
	For (each day) {
	Calculates the length of one time slot and Cycle according to Exps. (12) and (15), respectively
	}
Step 4. /* Detection Probability Calculation */	
	If (grid $\mathcal{G}_{(x,y)}$ is fully covered by the sensor s_i)
	Calculates $p(s_i, \mathcal{G}_{(x,y)})$ according to Exp. (16)
	else if (grid $\mathcal{G}_{(x,y)}$ is covered by the set of sensors $\Phi_{(x,y)}$)
	Calculates $p^{co}(\Phi_{(x,y)}, \mathcal{G}_{(x,y)})$ according to Exp. (17)
Step 5. /* Selecting the Set of Target Grid */	
	For (each grid $\mathcal{G}_{(1,j)}$ locates in the leftmost column of \mathbf{R}) {
	Select the first target grid \mathcal{G}_1^{tar} according to Exp. (20)
	}
	For $i=1: (c_2-1)$ {
	Determines the set of right neighboring grid $N_i(\mathcal{G}_{(x,y)})$
	Select the next target grid \mathcal{G}_{i+1}^{tar} according to Exp. (22)
	}
Step 6. /* Identification of the "Bottleneck Space-Time Point" */	
	For (each unscheduled sensor s_i) {
	Establish received schedules matrix $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$ according to Exp. (24)
	Calculates the surveillance quality matrix \mathcal{M}_i^{QoS} based on $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$
	Identifies the local bottleneck space-time point $o_{i,loc}^{weak}$ according to Exp. (19)
	}

Step 7. /* Constructing the Best Schedule for Each Sensor*/

For (each round of constructing the task schedule) {
For (each possible task schedule s_i^{sj} of s_j {
 Calculates the updated local surveillance quality matrix $\mathcal{M}_i^{QoS}(s_i^{sj})$.
 Identifies the new local space-time point $o_{i,loc}^{weak}(s_i^{sj})$ according to Exp. (27)
 selects the best task schedule $s_{best}^{s_i}$ according to Exps. (29) and (31).
 }
 }
 }

Step 8. /* The Random Back-off and Decision-Announcement */

For (each s_i)
 Calculates the back-off time BR_i according to Exp. (32)
 s_i countdown from BR_i to 0
If (there is no schedule broadcasted from $\mathbb{N}(s_i)$)
 s_i broadcasts its task schedule $s_{best}^{s_i}$
Else
 s_i receives schedule broadcasted from $\mathbb{N}(s_i)$ and $\mathcal{M}_{\mathbb{N}(s_i)}^{sch}$ updates
 Executes the operations from Step 8 to step 9 again;
Until all the sensors are scheduled

5 Simulation

This section studies the performance improvement of the proposed *BCRAS* algorithm against the existing studies *MCDP* [20] and *TCSAR* [14]. The *MCDP* adopts fixed sensing radius sensors to maximize the surveillance quality of the constructed barrier in *WRSNs*. The *TCSAR* further employs adjustable sensing sensors in *WRSNs* to maximize the surveillance quality of all points of interest within the monitoring area. This algorithm initially activates all sensors at their maximum sensing radius, then reduces the sensing radius of some sensors to further improve the detection probability at the weakest points of interest. Although *TCSAR* is initially designed as a target coverage algorithm, it can also be applied to barrier coverage if the target grids are considered as points of interest. Neither the *TCSAR* nor the *MCDP* algorithm considers the impact of different meteorological scenarios on *PV* power. Both of these algorithms assume that *PV* power is constant. The following presents the simulation environment and simulation results.

5.1 Simulation Environment

The simulation parameters are listed in Table 3. The size of the monitoring region is 1000m×40m. The grid size ranges from 2m to 10m. The number of deployed sensors varies from 200 to 1000, which are randomly deployed in the monitoring region. The minimum sensing radius of each sensor is set to 10m. The maximum sensing radius of each sensor is set to 30m, and the communication radius is twice the maximum sensing radius. The battery capacity of each sensor is 10.3 KJ [26], and the maximum power consumption rate of each sensor is 0.2J/s. The historical daily *PV* power data and corresponding meteorological data are from the dataset of the Desert Knowledge Australia (*DKA*) solar center. This dataset provides high-quality data related to *PV* power and corresponding

meteorological data in Northern Australia, which is an authoritative dataset widely used in the analysis and prediction of *PV* power [21].

Table 3. Simulation parameters

Parameter	Value
Simulation platform	Matlab R2024a
Monitoring region	1000m×40m
Number of sensors	200-1000
Grid size	2m-10m
Minimum sensing radius	10m
Maximum sensing radius	30m
Communication radius	60m
Battery capacity of each sensor	10.3 KJ
Maximum power consumption rate	0.2 J/s
<i>PV</i> power data	Dataset from <i>DKA</i> solar center
Deployment strategy	Randomly

5.2 Experiment Results

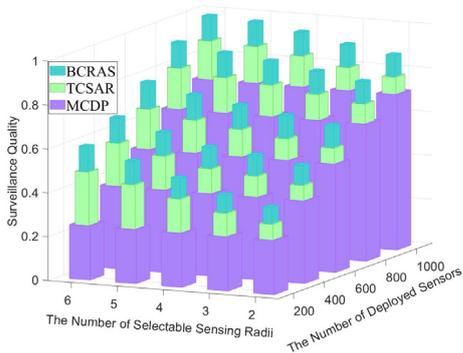
It is noted that weather conditions are generally divided into three main categories: clear, cloudy, and rainy days. This classification has been validated through the analysis of the dataset. Subsequently, Figure 8 compares the surveillance quality of three compared algorithms on clear, cloudy, and rainy days. The number of deployed sensors and the selectable sensing radius vary from 200 to 1000 and 2 to 6, respectively. The grid size is set to 6m. Because the *MCDP* algorithm adopts the fixed sensing radius sensor, it selects the maximum sensing radius as the sensing radius for each sensor in this simulation.

In general, the surveillance qualities of the three compared algorithms have a similar trend in that they are increased with the number of deployed sensors. This occurs because more sensors can be activated in each time slot, leading to a high surveillance quality. The other trend is the surveillance qualities of *BCRAS* and *TCSAR* are increased with the number of selectable sensing radii. The reason is that sensors with a larger number of selectable sensing radii enable a more refined partitioning of detection capabilities, thereby allowing for a more precise allocation of detection capabilities to bottleneck space-time points. This improves the utilization of each sensor's detection capability, resulting in improved surveillance quality. In addition, since the *MCDP* adopts fixed sensing radius sensors, the surveillance quality does not change with the varying number of selectable sensing radii. Furthermore, the surveillance qualities of the three algorithms are better on clear days compared to cloudy days and rainy days. This is because the solar-powered sensor can harvest more energy on a clear day. Consequently, each sensor can be activated in more time slots per cycle, which improves the surveillance quality.

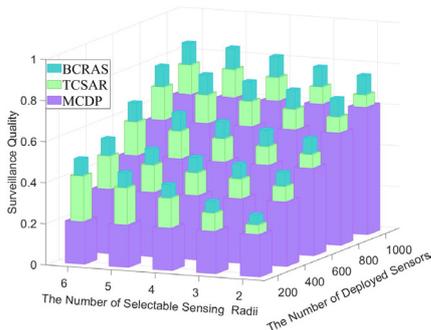
In comparison, the proposed *BCRAS* algorithm outperforms the *TCSAR* and *MCDP* in terms of surveillance quality in all cases. This occurs because the *BCRAS* algorithm considers the variable and weather-dependent nature of solar energy. It employs the CNN-LSTM model to predict the *PV* power for the next day accurately, such that the available energy from solar power of the next day can be evaluated precisely. The scheduling

scheme of each sensor is based on the evaluated available solar energy, which supports higher detection capacity while avoiding power depletion. In contrast, both *TCSAR* and *MCDP* assume that the *PV* power is constant. Under this assumption, the evaluation of *PV* power might be overly conservative to prevent some sensors from becoming non-operational due to power depletion.

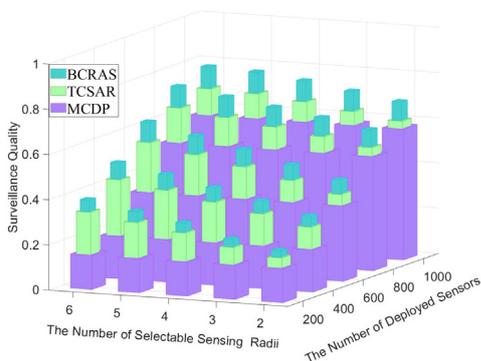
This experiment demonstrates Contribution 4 of this paper. By evaluating performance under different numbers of selectable sensing radii, we verify that dynamically adjusting the sensing radius improves the utilization of redundant detection capability and leads to higher surveillance quality, especially compared to fixed-radius schemes (*MCDP*).



(a) On clear days



(b) On cloudy days

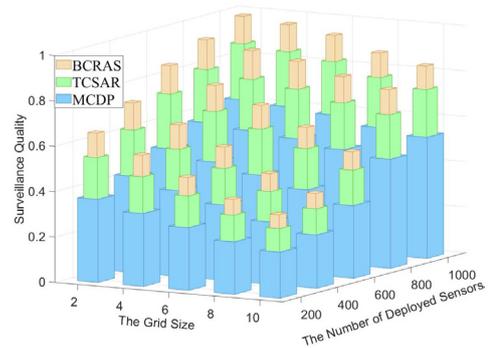


(c) On rainy days

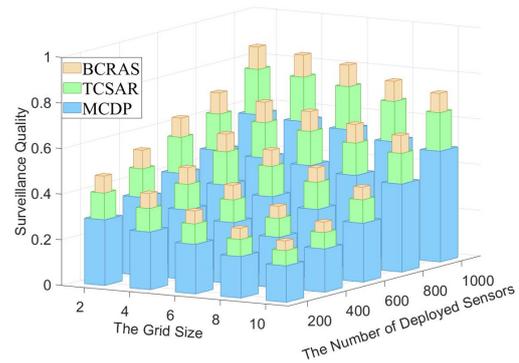
Figure 8. Comparison of *BCRAS*, *TCSAR*, and *MCDP* in terms of surveillance quality with varying numbers of deployed sensors and selectable sensing radii

Figure 9 further compares the three compared algorithms in terms of surveillance quality by varying the number of deployed sensors and the grid size on clear,

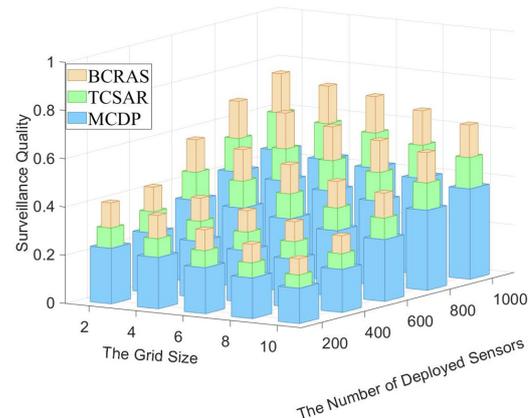
cloudy, and rainy days. The number of selectable sensing radii is set to 4. The common trend, similar to the results shown in Figure 8, is that the surveillance quality of the three algorithms increases with the number of deployed sensors, and the surveillance quality obtained by the three algorithms on clear days is higher than that on cloudy and rainy days. Besides, the surveillance qualities of the three algorithms decrease with the grid size increase. This occurs because more sensors are required to fully cover one grid with the larger size and hence leads to lower surveillance quality when the grid size grows.



(a) On clear days



(b) On cloudy days



(c) On rainy days

Figure 9. Comparison of *BCRAS*, *TCSAR*, and *MCDP* in terms of surveillance quality with varying numbers of deployed sensors and the grid size

In comparison, the proposed *BCRAS* algorithm achieves the best performance in terms of surveillance quality. This occurs because, in contrast to *TCSAR*, which

initially schedules all sensors with the maximum sensing radius and subsequently reduces the sensing radius of some sensors to enhance surveillance quality, *BCRAS* adjusts the sensing radius for each sensor during the task schedule design to maximize surveillance quality. This scheme allows *BCRAS* to exploit more opportunities for enhancing surveillance quality to space-time points. Additionally, *MCDP* adopts sensors with fixed sensing radii, which reduces the utilization of detection capability. As a result, the surveillance quality drops greatly as compared to that of *BCRAS*.

This experiment further supports Contribution 4. It illustrates how adaptive adjustment of sensing radii enables more efficient coverage, especially under finer grid resolutions. The *BCRAS* algorithm shows superior adaptability in adjusting detection patterns to maximize surveillance quality.

Figure 10 depicts the comparison of *BCRAS* and *TCSAR* in terms of the utilization of solar power under clear, cloudy, and rainy conditions. Since both *TCSAR* and *MCDP* algorithms assume that *PV* power is constant, they have identical solar energy utilization. Therefore, *MCDP* is not included in the comparisons in this experiment. Figures 10(a), 10(b), and 10(c) detail the real available energy (*RAE*) under clear, cloudy, and rainy days, respectively. Additionally, these figures compare the total consumed energy (*TCE*) and utilization of solar energy (*USE*) of the two compared algorithms under the same conditions, with each of Figures. 10(a), 10(b), and 10(c) compare ten days selected from respective weather conditions. Because the *TCSAR* assumes that *PV* power is constant and schedules sensors based on the constraint that charger energy equals consumed energy, it does not consider the variation of *PV* power and the effects of weather conditions. Therefore, the *TCSAR* adopts the average *PV* power of each month as the daily charging rate of the sensors for that same month. The value of *USE* can be calculated by applying the following expression.

$$USE = \frac{TCE}{RAE} \quad (36)$$

It should be noted that *USE* reaches 100% when *TCE* is equal to the *RAE*, which is the optimal utilization of solar energy (*OUSE*). In Figure 10, the *USE* less than 100% reflects that the total energy consumed by each sensor in a day exceeds the total energy obtained from solar power, posing a risk of sensor failure due to battery depletion. The *USE* larger than 100% reflects that the total energy consumed by each sensor in a day is less than the total energy that can be obtained from solar power, leading to a waste of solar power and consequently limiting the performance of the network. Therefore, the closer the *USE* is to the *OUSE*, the better the performance in terms of solar power utilization.

As shown in Figure 10, the solar power utilization of *BCRAS* achieves the best performance under rainy conditions. This occurs because the *PV* power under rainy conditions is relatively low and exhibits less variation.

The solar power utilization of *BCRAS* achieves the worst performance under cloudy conditions. This occurs because *PV* power under cloudy conditions is influenced by random cloud cover and is relatively more unstable compared to rainy and cloudy conditions, leading to higher discrepancies between the actual and estimated solar energy. However, overall, the performance of *BCRAS* is better than that of *TCSAR* in all cases. This is because the proposed *BCRAS* utilizes the CNN-LSTM model to predict future *PV* power, enabling the evaluation of available solar energy in advance. The *TCSAR* assumes the *PV* power is constant throughout each day, which will lead to a significant discrepancy between real solar energy and the estimated solar energy, that is the discrepancies between *RAE* and *TCE* of *TCSAR*. Therefore, the barrier coverage formed by the *TCSAR* poses a significant risk of sensor failure due to battery depletion on rainy days and results in a substantial waste of solar power on clear and cloudy days.

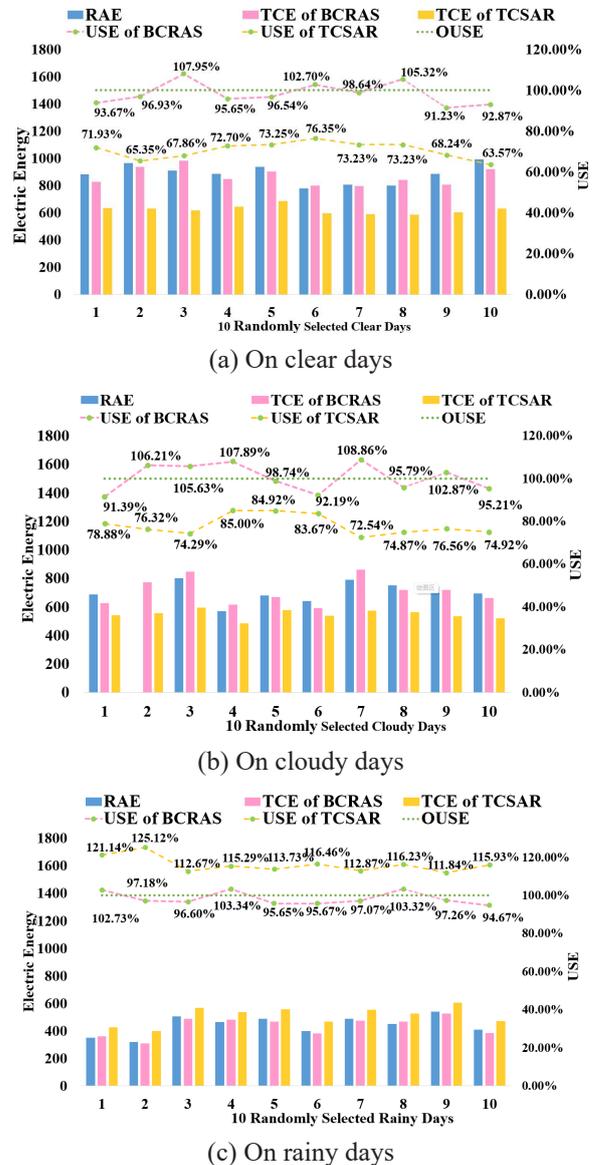


Figure 10. Comparison of *BCRAS* and *TCSAR* in terms of utilization of solar energy

This experiment demonstrates Contribution 1 and Contribution 2 of this paper. By comparing *BCRAS* with *TCSAR*, we show that the use of *CNN-LSTM*-based solar energy prediction and energy-balanced scheduling significantly improves solar power utilization and reduces the risk of battery depletion.

Figure 11 further compares the impact of path loss on the overall performance of *BCRAS*. Recall that the objective function of this study, the overall performance of *BCRAS* is surveillance quality of the barrier coverage achieved by *BCRAS*. In this experiment, the percentage degradation of surveillance quality (*PDSQ*) is adopted to reflect the impact of path loss on the overall performance of *BCRAS*, which is defined in Exp. (37).

$$PDSQ = \left(1 - \frac{q^{\lambda=0.001, \gamma=1}(BCRAS)}{q^{\lambda=x, \gamma=y}(BCRAS)} \right) \times 100\% \quad (37)$$

where λ and γ are the path loss exponents of the sensing signal strength of each sensor, $q^{\lambda=x, \gamma=y}$ is the surveillance quality achieved by the *BCRAS* algorithm when $\lambda = x$, $\gamma = y$, respectively. It is noted that $\lambda = 0.001$, $\gamma = 1$ represent the ideal condition, that is, the path loss is at the theoretical minimum value.

In this experiment, the number of deployed sensors is 600, the number of selectable sensing radii is 4, and the grid size is 6m. As shown in Figure 11, the value of *PDSQ* increases with higher values of the λ and γ , indicating a greater performance degradation under stronger path loss. This occurs because the higher value of λ leads to a greater rate of signal attenuation with distance. As path loss increases, the effective sensing radii of each sensor decreases, reducing the detection probability and overall surveillance quality of the barrier. Similarly, an increase in λ shapes the attenuation function that further accelerates the reduction of signal strength, contributing to a faster decay in detection probability and further decreasing monitoring effectiveness.

Furthermore, the simulation results show that path loss impacts performance more under certain weather conditions. On clear days, *PDSQ* increases less due to better solar energy collection, enabling larger sensing radii and longer sensing operations. In contrast, rainy or cloudy days reduce solar energy, shortening sensing radii and increasing *PDSQ*.

This experiment reflects Contribution 3 of this paper. It evaluates how cooperative detection performance degrades under different path loss conditions, showing that the *BCRAS* algorithm's strategy of prioritizing high-contribution sensors effectively mitigates degradation.

Finally, to investigate the effectiveness of the Farthest Distance Scheduling First (*FDSF*) policy, Figure 12 further illustrates the comparison of sensor scheduling strategies, with and without the application of the *FDSF* policy, by presenting the cooperative detection probabilities at 54 space-time points, which correspond to 9 selected target grids and 6 time slots in one selected cycle. Figure 12(a) and Figure 12(b) present the experiment result of sensor scheduling strategies without and with applying the *FDSF*

policy. By comparing the experiment results of Figure 12, the sensor scheduling strategy by applying the *FDSF* policy outperforms the one without applying the *FDSF* policy in terms of surveillance stability and surveillance quality. This occurs because the greater the distance between a sensor and the target grid, the fewer selectable sensing radii are available that can encompass the target grid, thereby reducing the chances of covering bottleneck space-time points. Without employing the *FDSF* policy, these sensors, which are distant from the target grid, may not contribute to improving the surveillance quality at bottleneck space-time points. Conversely, sensors closer to the target grid can cover more target grids, thus having a greater selection of radius and more opportunities to enhance the monitoring quality at bottleneck points. Therefore, prioritizing sensors farther from the target grid in scheduling maximizes the use of all sensors' monitoring capabilities to cover more surveillance bottleneck points, thereby improving the overall surveillance quality and stability.

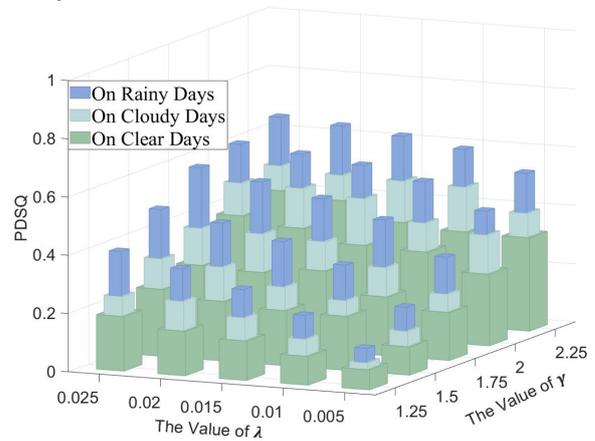
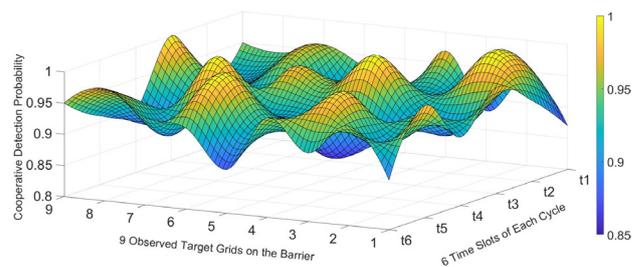
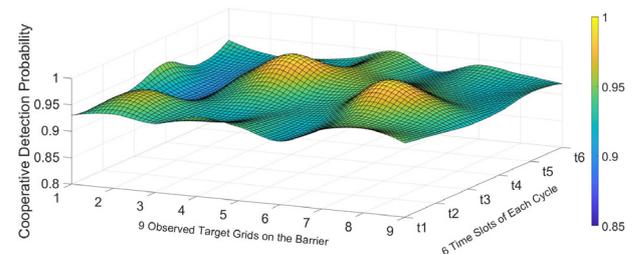


Figure 11. Performance comparison of the *BCRAS* algorithm with different values of λ and γ



(a) Without applying *FDSF* policy



(b) By applying *FDSF* policy

Figure 12. Comparisons of the cooperative detection probability at different space-time points with and without applying *FDSF* policy to sensor scheduling

This experiment also supports Contribution 3. By comparing the cooperative detection probabilities with and without the *FDSF* policy, the experiment confirms that prioritizing far-sensors enhances bottleneck space-time point coverage, thereby improving overall surveillance quality.

6 Conclusion

This paper proposes a novel barrier coverage algorithm for *WRSNs*, called *BCRAS*, which applies the adjustable sensing radii sensor and aims to maximize the surveillance quality while balancing the energy consumption and acquisition to perpetuate network lifetime. To achieve this objective, the CNN-LSTM model is applied to predict the *PV* power in advance. Based on the prediction of *PV* power, the *BCRAS* algorithm partitions the monitoring region into several identical grids and time slots, which form multiple space-time points, and allocates the same amount of energy for each sensor within each cycle. This allocation includes a reserve for nighttime sensor operation. Then *BCRAS* identified the local bottleneck space-time point of each sensor, subsequently constructing the best task schedule for each sensor, which includes the best sensing radius and corresponding activated time slot, aiming to maximize the surveillance quality while balancing the acquired and consumed energy of each sensor. Experimental results show that the proposed *BCRAS* outperforms the existing algorithms in terms of surveillance quality in different scenarios and the utilization of solar energy. Despite the promising results, this study has several limitations. It assumes homogeneous sensor nodes and relies on accurate meteorological data for *PV* power prediction, which may not hold in all real-world scenarios. In future work, we plan to consider heterogeneous sensor models and enhance the robustness of the prediction and scheduling strategies under uncertain environmental conditions.

References

- [1] H. E. Houssein, M. A. Othman, W. M. Mohamed, M. Younan, Internet of Things in Smart Cities: Comprehensive Review, Open Issues, and Challenges, *IEEE Internet of Things Journal*, Vol. 11, No. 21, pp. 34941–34952, November, 2024.
<https://doi.org/10.1109/JIOT.2024.3449753>
- [2] W. Osamy, A. M. Khedr, A. Salim, Design Workload Aware Data Collection Technique for IoT-enabled WSNs in Sustainable Smart Cities, *IEEE Transactions on Sustainable Computing*, Vol. 10, No. 2, pp. 244–261, March-April, 2025.
<https://doi.ieeecomputersociety.org/10.1109/TSUSC.2024.3418136>
- [3] S. Balaji, R. Pavithra, D. Arivudainambi, K. A. Varunkumar, A. Suresh, M. M. Omar, A. K. Bashir, Toward Efficient Sensor Deployment in Internet of Things for Target Coverage and Sensor Connectivity, *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 3, pp. 5432–5440, August, 2024.
<https://doi.org/10.1109/TCE.2023.3335466>
- [4] X. Zhu, J. Li, M. Zhou, Target Coverage-Oriented Deployment of Rechargeable Directional Sensor Networks With a Mobile Charger, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5196–5208, June, 2019.
<https://doi.org/10.1109/JIOT.2019.2899155>
- [5] S. He, J. Chen, Y. Shu, X. Cui, K. Shi, C. Wei, Z. Shi, Efficient Fault-Tolerant Information Barrier Coverage in Internet of Things, *IEEE Transactions on Wireless Communications*, Vol. 20, No. 12, pp. 7963–7976, December, 2021.
<https://doi.org/10.1109/TWC.2021.3089039>
- [6] H. M. Ammari, Investigating Physical Security in Stealthy Lattice Wireless Sensor Networks Using *k*-Barrier Coverage, *Ad Hoc Networks*, Vol. 89, pp. 142–160, June, 2019.
<https://doi.org/10.1016/j.adhoc.2018.11.003>
- [7] X. Zhang, Y. Zhou, Q. Zhang, V. C. S. Lee, M. Li, Problem Specific MOEA/D for Barrier Coverage with Wireless Sensors, *IEEE Transactions on Cybernetics*, Vol. 47, No. 11, pp. 3854–3865, November, 2017.
<https://doi.org/10.1109/TCYB.2016.2585745>
- [8] C.-I. Weng, C.-Y. Chang, C.-Y. Hsiao, C.-T. Chang, H. Chen, On-Supporting Energy Balanced *k*-Barrier Coverage in Wireless Sensor Networks, *IEEE Access*, Vol. 6, pp. 13261–13274, January, 2018.
<https://doi.org/10.1109/ACCESS.2018.2792678>
- [9] P. Si, J. Ma, F. Tao, Z. Fu, L. Shu, Energy-Efficient Barrier Coverage with Probabilistic Sensors in Wireless Sensor Networks, *IEEE Sensors Journal*, Vol. 20, No. 10, pp. 5624–5633, May, 2020.
<https://doi.org/10.1109/JSEN.2020.2970435>
- [10] V. K. Akram, O. Dagdeviren, B. Tavli, A Coverage-Aware Distributed *k*-Connectivity Maintenance Algorithm for Arbitrarily Large *k* in Mobile Sensor Networks, *IEEE/ACM Transactions on Networking*, Vol. 30, No. 1, pp. 62–75, February, 2022.
<https://doi.org/10.1109/TNET.2021.3104356>
- [11] D. Shishika, D. G. Macharet, B. M. Sadler, V. Kumar, Game Theoretic Formation Design for Probabilistic Barrier Coverage, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, 2020, pp. 11703–11709.
<https://doi.org/10.1109/IROS45743.2020.9340724>
- [12] X. Fan, F. Hu, T. Liu, K. Chi, J. Xu, Cost Effective Directional Barrier Construction Based on Zooming and United Probabilistic Detection, *IEEE Transactions on Mobile Computing*, Vol. 19, No. 7, pp. 1555–1569, July, 2020.
<https://doi.org/10.1109/TMC.2019.2912749>
- [13] Z. G. Chen, Y. Lin, Y. J. Gong, Z. H. Zhan, J. Zhang, Maximizing Lifetime of Range-Adjustable Wireless Sensor Networks: A Neighborhood-Based Estimation of Distribution Algorithm, *IEEE Transactions on Cybernetics*, Vol. 51, No. 11, pp. 5433–5444, November, 2021.
<https://doi.org/10.1109/TCYB.2020.2977858>
- [14] Q. Zhang, C. Y. Chang, Z. Dong, D. S. Roy, TCSAR: Target Coverage Mechanism for Sensors With Adjustable Sensing Range in WRSNs, *IEEE Sensors Journal*, Vol. 22, No. 4, pp. 3756–3765, February, 2022.
<https://doi.org/10.1109/JSEN.2021.3139731>
- [15] Z. Zhang, W. Wu, J. Yuan, D. Z. Du, Breach-Free Sleep-Wakeup Scheduling for Barrier Coverage With Heterogeneous Wireless Sensors, *IEEE/ACM Transactions on Networking*, Vol. 26, No. 5, pp. 2404–2413, October, 2018.
<https://doi.org/10.1109/TNET.2018.2867156>
- [16] N. T. Hanh, H. T. T. Binh, N. V. Son, N. T. Trang, P.

- N. Lan, Optimizing Wireless Sensor Network Lifetime Through K-Coverage Maximization and Memetic Search, *Sustainable Computing: Informatics and Systems*, Vol. 40, Article No. 100905, December, 2023.
<https://doi.org/10.1016/j.suscom.2023.100905>
- [17] D. Thomas, R. Shankaran, M. A. Orgun, S. C. Mukhopadhyay, SEC2: A Secure and Energy Efficient Barrier Coverage Scheduling for WSN-Based IoT Applications, *IEEE Transactions on Green Communications and Networking*, Vol. 5, No. 2, pp. 622–634, June, 2021.
<https://doi.org/10.1109/TGCN.2021.3067606>
- [18] W. Wen, C. Y. Chang, Q. Zhang, D. S. Roy, PLPR: Joint Perpetual Lifetime and Path Reduction for Energy Recharging Using Multiple Mobile Chargers in WRSNs, *IEEE Sensors Journal*, Vol. 24, No. 19, pp. 31157–31171, October, 2024.
<https://doi.org/10.1109/JSEN.2024.3434374>
- [19] C. Shang, C. Y. Chang, W. H. Liao, D. S. Roy, RLR: Joint Reinforcement Learning and Attraction Reward for Mobile Charger in Wireless Rechargeable Sensor Networks, *IEEE Internet of Things Journal*, Vol. 10, No. 18, pp. 16107–16120, September, 2023.
<https://doi.org/10.1109/JIOT.2023.3267242>
- [20] P. Xu, J. Wu, C. Y. Chang, C. Shang, D. S. Roy, MCDP: Maximizing Cooperative Detection Probability for Barrier Coverage in Rechargeable Wireless Sensor Networks, *IEEE Sensors Journal*, Vol. 21, No. 5, pp. 7080–7092, March, 2021.
<https://doi.org/10.1109/JSEN.2020.3043456>
- [21] S. Zheng, D. Li, Y. Li, Z. Zeng, Y. Zhou, C. Li, K. Song, An Adaptive Weighted Stacking Ensemble Framework for Photovoltaic Power Generation Forecasting with Joint Optimization of Features and Hyperparameters, *Engineering Applications of Artificial Intelligence*, Vol. 144, Article No. 110075, March, 2025.
<https://doi.org/10.1016/j.engappai.2025.110075>
- [22] S. Silvestri, K. Goss, MobiBar: An Autonomous Deployment Algorithm for Barrier Coverage with Mobile Sensors, *Ad Hoc Networks*, Vol. 54, pp. 111–129, January, 2017.
<https://doi.org/10.1016/j.adhoc.2016.10.010>
- [23] H. P. Gupta, S. V. Rao, T. Venkatesh, Sleep Scheduling Protocol for k-Coverage of Three-Dimensional Heterogeneous WSNs, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 10, pp. 8423–8431, October, 2016.
<https://doi.org/10.1109/TVT.2015.2508801>
- [24] A. Rossi, A. Singh, M. Sevaux, An Exact Approach for Maximizing the Lifetime of Sensor Networks with Adjustable Sensing Ranges, *Computers & Operations Research*, Vol. 39, No. 12, pp. 3166–3176, December, 2012.
<https://doi.org/10.1016/j.cor.2012.04.001>
- [25] Q. Li, N. Zhang, M. Cheffena, X. Shen, Channel-Based Optimal Back-Off Delay Control in Delay-Constrained Industrial WSNs, *IEEE Transactions on Wireless Communications*, Vol. 19, No. 1, pp. 696–711, January, 2020.
<https://doi.org/10.1109/TWC.2019.2948156>
- [26] C. Wang, J. Li, F. Ye, Y. Yang, A Mobile Data Gathering Framework for Wireless Rechargeable Sensor Networks with Vehicle Movement Costs and Capacity Constraints, *IEEE Transactions on Computers*, Vol. 65, No. 8, pp. 2411–2427, August, 2016.
<https://doi.org/10.1109/TC.2015.2490060>
- [27] Q. Yu, C. Yang, G. Dai, L. Peng, J. Li, A Novel Penalty Function-Based Interval Constrained Multi-Objective Optimization Algorithm for Uncertain Problems, *Swarm and Evolutionary Computation*, Vol. 88, Article No. 101584, July, 2024.
<https://doi.org/10.1016/j.swevo.2024.101584>
- [28] M. A. Houran, S. M. S. Bukhari, M. H. Zafar, M. Mansoor, W. Chen, COA-CNN-LSTM: Coati Optimization Algorithm-Based Hybrid Deep Learning Model for PV/Wind Power Forecasting in Smart Grid Applications, *Applied Energy*, Vol. 349, Article No. 121638, November, 2023.
<https://doi.org/10.1016/j.apenergy.2023.121638>
- [29] D. A. Guimaraes, E. P. Frigieri, L. J. Sakai, Influence of Node Mobility, Recharge, and Path Loss on the Optimized Lifetime of Wireless Rechargeable Sensor Networks, *Ad Hoc Networks*, Vol. 97, Article No. 102025, February, 2020.
<https://doi.org/10.1016/j.adhoc.2019.102025>
- [30] S. M. J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, S. Nahavandi, Automated Deep CNN-LSTM Architecture Design for Solar Irradiance Forecasting, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 1, pp. 54–65, January, 2022.
<https://doi.org/10.1109/TSMC.2021.3093519>
- [31] W.-Y. Chang, P. Soma, H. Chen, H. Chang, C.-W. Tsai, A hybrid firefly with dynamic multi-swarm particle swarm optimization for WSN deployment, *Journal of Internet Technology*, Vol. 24, No. 4, pp. 825–836, July, 2023.
<https://doi.org/10.53106/160792642023072404001>
- [32] Y. Kan, C.-Y. Chang, S.-J. Wu, C.-H. Kuo, D. S. Roy, CSCP: energy charging mechanism for surveillance quality, network connectivity and perpetual lifetime in WRSNs, *Journal of Internet Technology*, Vol. 24, No. 2, pp. 461–473, March, 2023.
<https://doi.org/10.53106/160792642023032402023>
- [33] H. Yu, C.-Y. Chang, W. Guan, Y. Wang, D. S. Roy, MC³: A Multicharger Cooperative Charging Mechanism for Maximizing Surveillance Quality in WRSNs, *IEEE Sensors Journal*, Vol. 24, No. 10, pp. 17080–17091, May, 2024.
<https://doi.org/10.1109/JSEN.2024.3382887>

Biographies



Pei Xu received the Ph.D. degree in Computer Science and Technology from Anhui University, Hefei, China, in 2020. He is currently a Lecturer with the School of Artificial Intelligence and Big Data, Hefei University, Hefei, China. His current research interests include smart sensing technology, the Internet of Things, and deep learning.



Youxi Li received the Ph.D. degree in computer science and information engineering from Tamkang University, New Taipei, Taiwan, in 2024. His current research interests are the Internet of Things, data analysis and artificial intelligence.



Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from the National Central University, Taiwan, in 1995. He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the

Department of Artificial Intelligence, Tamkang University, New Taipei, Taiwan. His current research interests include artificial intelligence, deep learning and machine learning, the Internet of Things, and wireless sensor networks. He has been serves as an associate guest editor for several SCI-indexed journals, including International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC) since 2011, Journal of Applied Science and Engineering (JASE) since 2018, International Journal of Distributed Sensor Networks (IJDSN) from 2012 to 2014, IET Communications in 2011, Telecommunication Systems (TS) in 2010, Journal of Information Science and Engineering (JISE) in 2008, and Journal of Internet Technology (JIT) from 2004 to 2008. Dr. Chang is the Co-Chair of ACM SIGMOBILE, Taiwan.



Chin-Hwa Kuo received the B.S. degree in mechanical engineering from Chung-Yuan University, Taoyuan City, Taiwan, in 1980, the M.S. degree in electrical engineering from Marquette University, Milwaukee, WI, USA, in 1989, and the Ph.D. degree in electrical engineering from the University of Notre

Dame, South Bend, IN, USA, in 1994. He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His research interests include multimedia processing and applications, learning analytics, and large-scale deployment of information communications technology in education.



Diptendu Sinha Roy received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010. In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, India, as an Associate Professor, where he has

been working as the Chair of the Department of Computer Science and Engineering, since January 2017. Prior to his stint at NIT Meghalaya, he worked with the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed & cloud computing, and the IoT, specifically applications of artificial intelligence/machine learning for smart integrated systems. Dr. Sinha Roy is a member of the IEEE Computer Society.