ARTICLE IN PRESS

Journal of Engineering Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Engineering Research

journal homepage: www.journals.elsevier.com/journal-of-engineering-research

An energy efficient clustering algorithm based on density and fitness for mobile crowd-sensing network

Hao Li [a,1], Hongwei Wang [b,1], Kaiyu Wang [a], Tonghui Qu [c], Xunhuan Ren [a], Jun Ma [a,*]

[a] Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
[b] Forestry School of materials Engineering Northeast Forestry University, HeiLongJiang 150040, China
[c] Hangzhou Hikvision Digital Technology Co., Ltd., Hangzhou 310052, China

ARTICLE INFO

Keywords:
Mobile crowd-sensing
Clustering algorithm
Energy efficient
Node density

ABSTRACT

Mobile crowd-sensing (MCS) is a cutting-edge paradigm that gathers sensory data and generates valuable insights for a multitude of users by utilizing built-in sensors and social applications in mobile devices. This enables a broad spectrum of Internet of Things (IoT) services. We introduce a novel MCS algorithm, Mobile Crowd-sensing Low Energy Clustering (MCLEC), which employs advanced clustering techniques to address issues of data oversampling and energy inefficiency prevalent in MCS networks. MCLEC innovatively adjusts clustering radii based on local node density and the proximity of nodes to the cloud server, thus optimizing data transmission paths and reducing energy consumption. A pivotal enhancement in MCLEC is its cluster head election strategy, which prioritizes leaders based on their energy levels and mobility, thereby enhancing network stability and minimizing the frequency of head re-elections. Our comparisons with established algorithms such as LEACH, LEACH-C, LEACH-M, DEEC, and SEP show that MCLEC significantly improves energy efficiency, reduces server load, and prolongs the lifespan of network nodes, establishing it as an effective solution for IoT applications dependent on MCS. Additionally, MCLEC was compared with other novel clustering algorithms including E-FLZSEPFCH, DFLC, ECPF, ACAWT, UCR, CHEF, and Gupta's algorithm. The results indicate that MCLEC also surpasses most of these algorithms in terms of energy consumption and network lifetime.

Introduction

Mobile crowd-sensing (MCS) application [1] is proposed as a new perceptual style of the Internet of Things when mobile wireless sensor technology and mobile sensing devices are ubiquitous. MCS application combines crowdsourcing ideology [2] and mobile wireless sensor networks and breaks the rules of local perception in traditional wireless networks. Through network awareness cooperation, it forms a MCS network, and completes the distribution of perception tasks, data collection, and processing, thus completing a large number of complex social sensing tasks [3].

Due to the diversity of mobile wireless sensors, the MCS network has achieved good performance in environmental monitoring [4], intelligent transportation [5], public security [6], and social services [7]. A variety of MCS technologies have been developed and designed in the research and industrial fields [8–10]. But they all have single data collection, unbalanced resource allocation, and excessive energy consumption. Whether based on the data collection method of the mobile convergence node or single-hop communication, the mobile sensing node directly communicates with the cloud. Directly uploading a large number of raw sensing data to the cloud for processing will not only cause load imbalance but also cause excessive energy consumption and a short network life cycle. At the same time, the transmission mode with high energy consumption will increase the energy consumption of the network, and shorten the life cycle of the network, which reduces the willingness of users to participate. Li et al. [11] introduced a clustering algorithm aimed at discovering routes on demand, with a specific focus on mitigating node energy consumption in unique circumstances.

Before the popularity of mobile wireless sensor networks (WSN), traditional static WSN [12] put forward the concept of clustering to expand network coverage, shorten transmission distance, and reduce network energy consumption. Clustering network structure has good scalability and robustness [13]. It can save energy, balance the load, and distribute resources reasonably. So clustering algorithm has become one

* Corresponding author.
E-mail address: majun@bsuir.by (J. Ma).
[1] Co-first author.

https://doi.org/10.1016/j.jer.2024.07.011
Received 5 January 2024; Received in revised form 24 May 2024; Accepted 16 July 2024
Available online 17 July 2024

Please cite this article as: Hao Li et al., *Journal of Engineering Research*, https://doi.org/10.1016/j.jer.2024.07.011

of the main methods to extend the life cycle of WSNs and reduce the energy consumption of nodes at home and abroad. In the study, the idea of clustering is proposed to reduce the energy consumption of the network. When a static clustering algorithm is applied to a mobile crowd-sensing network, it can not form a complete network topology quickly, resulting in the rapid energy consumption of nodes. LEACH-M, as a traditional static routing protocol over to mobile routing protocol, also has the problems of random cluster head (CH) election and uneven cluster head distribution. If the LEACH-M algorithm is directly introduced into a large-scale mobile crowd-sensing network, it will directly lead to a serious imbalance of network energy consumption and load, and cannot guarantee the stability of network performance. Therefore, the traditional clustering algorithm is not suitable for the MCS network.

Firstly, the network is divided into clusters. Before the cluster head election, it is necessary to determine which nodes are eligible to compete for the cluster head role. Two points need to be considered: (1) Due to the limited energy of mobile nodes, multi-link data transmission cannot be sustained for a long time. (2) It is impossible for all nodes to belong to the same cluster indefinitely.

To address these points, this paper proposes methods to solve these issues. The algorithm introduced here enhances the stability and longevity of Mobile Crowd-Sensing (MCS) networks by innovating the method for cluster head election. It considers not only the residual energy of potential cluster heads but also incorporates their relative velocity and duration within the cluster, optimizing both energy efficiency and the network's dynamic adaptability.

Additionally, the proposed cluster group member selection methodology addresses the limitations of traditional clustering protocols by dynamically adjusting the cluster radius based on node density and proximity to the cloud server. This method effectively manages data aggregation and minimizes network transmission energy, paving the way for more sustainable MCS network operations.

MCS background and related work

Background on MCS

The deployment of a Mobile Crowd-Sensing network mainly relies on individuals or mobile devices for carrying, so the equipment deployment in the presence of a large number of mobile facilities in the city will greatly save cost and time. In wireless communication between network nodes, the opportunistic forwarding mode of "storage-carriage-forwarding" [14] can effectively improve the data transmission effect and enhance the cooperation ability between nodes. As shown in Fig. 1, the typical Mobile Crowd-Sensing network structure is mainly composed of three components [15–17], which are: 1. Initiator of perception task, 2. Task distribution system, 3. Service node.

Due to the limited sensing ability of smart devices, the selfishness of human beings and the limited performance of devices also determine that people will not install too many sensors in mobile terminals to increase the energy supply pressure of devices and consume a lot of energy in the process of data collection and transmission. Therefore, in future development, the network cost will inevitably face the challenge of fast energy consumption and a short life cycle. The control of data perception cost mainly lies in the selection of users, while the mitigation of excessive energy consumption lies in the cooperative perception among participants. Both rely on proper task transfer and data aggregation. Literature [18] predicts user calls and movements, enabling users to report data in appropriate actual and network environments to reduce traffic consumption and power consumption. Literature [19] USES static users to assist information transmission to reduce energy consumption based on the Archimedes curve.

Network clustering protocol analysis

Traditional WSNs are composed of static nodes and controllable mobile nodes, which can provide reliable sensing coverage services. To expand its coverage and shorten the transmission distance of nodes, the clustering mechanism divides the whole network into three connected areas, and the node information is transmitted up one level after another. Most of the time, the member nodes shut down the communication module, and the cluster head contact center server is responsible for long-distance information transmission and routing forwarding. This improves communication coverage in the area while saving network energy and reducing network traffic.

LEACH routing protocol

LEACH routing protocol [20] is a classical clustering protocol for wireless sensor networks. It uses a "round" to randomly select cluster head nodes. When the random number generated by a node is less than the threshold, it is elected as a cluster head and broadcast to the entire network. Nodes within this radius are defined as affiliated nodes. This method achieves equal opportunities to serve as cluster heads for all nodes, which can balance the overall energy of the network.

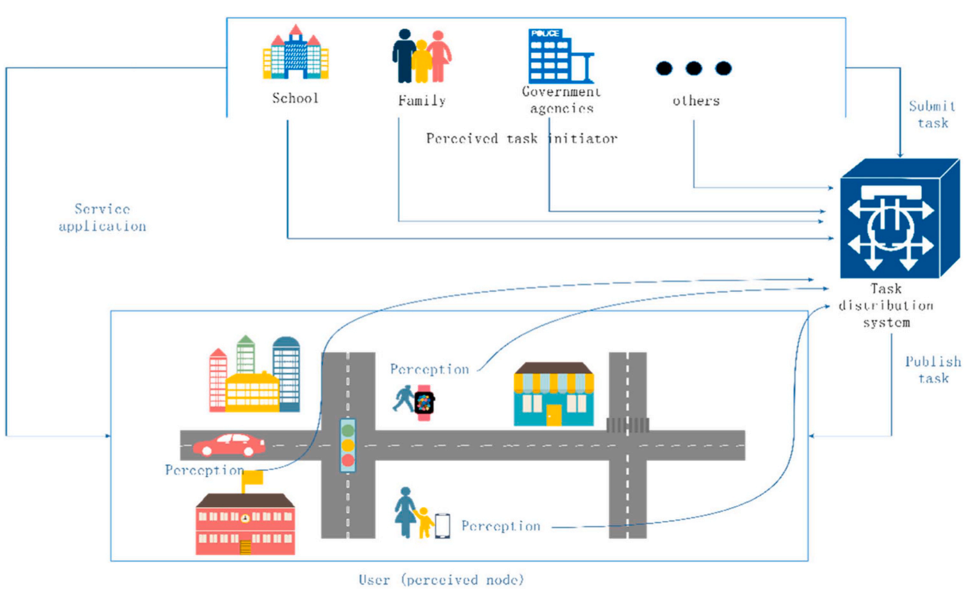

Fig. 1. Mobile crowd-sensing structure diagram.

The duration of the LEACH protocol in cluster construction is far less than that of a stable transmission node and cyclic random cluster head election mechanism, which greatly saves the energy consumption of the network. However, the LEACH protocol has many shortcomings. Therefore, based on the LEACH protocol, researchers improved it and gave rise to similar clustering protocols such as LEACH-C [21] and LEACH-M [22]. Each sensor node in the LEACH-C protocol includes its location and residual energy in the information it sends to the base station. The base station makes statistics on the collected data and obtains the average energy of the current network nodes. The nodes with higher than the average energy participate in clust-head election, and determine the number of clust-head according to the network environment. Then, the simulated annealing algorithm is used to find the network node with the best position as the cluster head.

LEACH-M adds a data response mechanism based on LEACH to realize the support of node mobility. In the stable working stage, the nodes in the cluster will move out of the signal coverage of the cluster head. For this situation, the solution adopted by the LEACH-M protocol is shown in Fig. 2. If the node cannot receive the request sent by the cluster head in the first designated time slot and does not receive the data request sent by the cluster head in the next time slot, it determines that it has moved out of the cluster and will send a message to other cluster head nodes. Broadcast and apply to join the cluster.

DEEC routing protocol

DEEC algorithm [14,23] is an efficient and efficient distributed clustering routing protocol suitable for multi-stage energy. The main principle is to use the ratio of the current residual energy to the average energy of the whole network to realize the election of cluster heads, the energy balance, and the improvement of the whole network life cycle. Once the cluster head is determined, it will start to broadcast its ID information to the surrounding and select the nearest node to the cluster head to join the cluster and become a member of the cluster, thus forming a cluster network.

Advanced and optimized WSN clustering techniques

In recent advancements in Wireless Sensor Networks (WSNs), researchers have shifted focus towards more sophisticated and adaptable strategies to address the limitations of traditional protocols such as LEACH and DEEC. Unlike LEACH, which primarily relies on randomized cluster head rotations, and DEEC, which focuses on energy efficiency through a simple energy ratio mechanism, the latest studies incorporate dynamic algorithms, strategic deployment, and innovative optimization techniques. These contemporary approaches aim to enhance network performance, improve security measures, and optimize resource management in increasingly complex network environments.

Researchers in [24] developed a dynamic fuzzy-based clustering protocol tailored for cognitive radio wireless sensor networks (CR-WSNs). In [25], a novel approach using meta-heuristic algorithms was introduced to enhance clustering and routing in IoT-based wireless sensor networks. The study by [26] introduces a multi-attribute decision-making framework for optimal deployment of sensor nodes in WSNs. In [27], researchers explored optimized clustering and load balancing in wireless sensor networks using a multi-attribute decision-making approach. The research in [28] proposes a strategic node placement method that utilizes partition-based techniques to improve communication efficiency in WSNs. In [29], the paper proposes an enhanced whale optimization algorithm for node localization in wireless sensor networks, focusing on metrics like localization error rate and convergence rate. The study in [30] discusses the application of a simplified, energy-efficient blockchain implementation for cognitive wireless communication networks. In [31], researchers propose a fuzzy-based clustering algorithm that aims to enhance node coverage and load balancing in Wireless Sensor Networks. This method integrates multiple conflicting factors, optimizing network lifetime and ensuring balanced energy consumption among nodes through a fuzzy logic approach. The paper in [32] introduces a matrix method to improve the efficiency of non-dominated sorting and population selection in multi-objective optimization problems. This approach reduces computational complexity and accelerates the convergence process, offering a practical alternative to traditional evolutionary algorithms. Researchers in [33] developed an opportunistic energy-efficient dynamic self-configuration routing algorithm for IoT applications using WSNs. This novel algorithm dynamically configures clusters based on nodes' residual energy and mobility, enhancing key performance metrics such as throughput, delay, and packet delivery ratio. Researchers in [34] enhanced the fuzzy logic zone stable election protocol for cluster head election, combined with multipath routing to enhance wireless sensor network efficiency. This approach outperforms traditional protocols like LEACH by optimizing cluster head selection and ensuring more reliable data transmission, reducing energy consumption, delay, and packet loss significantly.

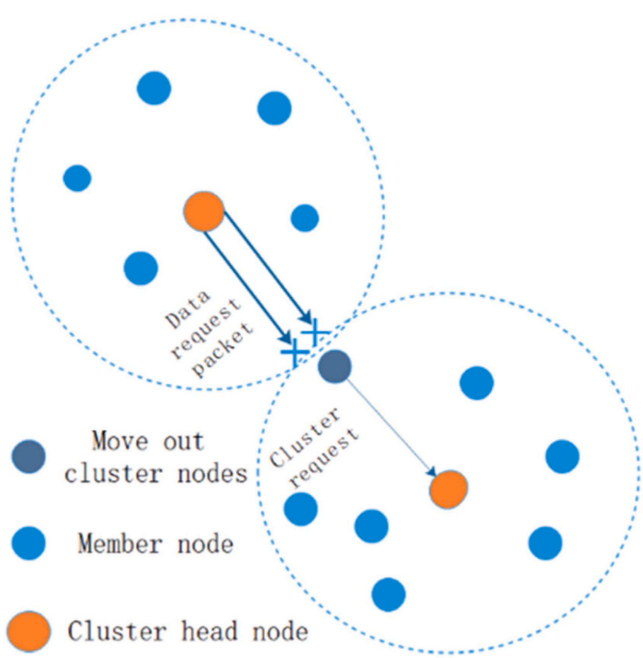

Fig. 2. LEACH-M protocol node interaction.

Disadvantages of clustering protocols

Although the development of routing clustering protocols in wireless sensor networks has been gradually mature, each clustering algorithm has some shortcomings in the application environment. It needs to be improved as follows:

(1) Unreasonable cluster head election. Most algorithms in clustering protocol adopt dynamic random or comparison thresholds to select cluster heads. The random method may cause premature death of cluster heads due to too low energy. The disadvantage of the threshold is that if the data that does not meet the threshold standard cannot be transmitted, the sink node may be unable to obtain any data information.
(2) High additional energy consumption. The cluster radius division of most clustering protocols depends on the communication radius of the cluster-head node. If the density of the cluster-head node is relatively low and the nodes are clustered around the cluster-head, the excessive transmission radius will consume the energy of the cluster-head too quickly and seriously waste the limited energy resources in the wireless sensor network.

Based on fitness and node density clustering mechanism in MCS

Data perception, monitoring, and fusion are intricate processes within the MCS (Mobile Crowd-Sensing) network, especially in intermittently connected network environments where an opportunistic forwarding mechanism is necessary to transmit perceptual data. Transmitting large volumes of raw perceptual data can lead to significant transmission delays, which may hinder the incentive mechanisms aimed at recruiting perceptual users. Moreover, processing such extensive data directly within the MCS network consumes considerable energy and network resources, placing undue stress on servers.

As the capabilities of intelligent terminals improve markedly, there is a shift towards networks that are low in both transmission energy consumption and computing costs, aligning with the energy-efficient characteristics of clustering algorithms. This transition supports the cost optimization in Crowd-Sensing networks. Furthermore, long-distance transmissions between the cluster head and the cloud server are more resource-efficient and effective at reducing network congestion than transmissions from each individual node to the server. As illustrated in Fig. 3, the clustering transmission mode significantly reduces the number of feedback links within the sensing region compared to traditional transmission modes.

In our algorithm, we define assumptions to support the correct working of the MCLEC algorithm:

- Sensor nodes move irregularly.
- Sensor nodes can obtain their current position.
- All nodes have the same data processing and communication capabilities.
- The energy of the sensor is limited, but if the energy reaches a certain value, the node will die, that is, it will no longer sense data.
- Sensor nodes use the same transmit power and range.
- Cluster head nodes actively and periodically send data to the base station.
- Cloud servers are always in a static state.

Initiation of the process commences with the calculation of node density, denoted as $\rho(N_i)$, which serves as a foundational parameter for determining the dynamic clustering radius R. This radius is pivotal in facilitating the election of cluster heads, ensuring that only the most suitably positioned nodes, in terms of connectivity and energy resources, are selected for this critical role.

Following the election of the cluster head, the algorithm evaluates the necessity for re-election of cluster heads through a decision-making node, which addresses potential shifts in node distribution or alterations in energy levels that may compromise data transmission efficiency or network stability. As shown as the Fig. 4.

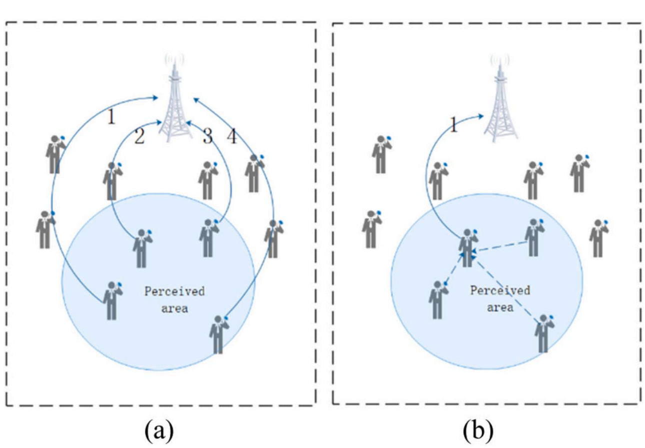

Fig. 3. A comparison of the two data transmission modes. (a) Data transmission by traditional group perceptive network. (b) Data transmission by cluster group perceptive network.

In the stage of cluster perception and data transmission, cluster division and cluster head election are completed. When the service node is in these two phases, the state is mainly divided into discovery, work, and sleep. When the node is in the discovery stage, it first perceives the surrounding data, elects the cluster head, and the non-cluster-head node chooses its cluster group, issues a request to the cluster head, and stops the data transmission. When the data collection and election of the cluster head are completed, the node will switch to the working state. At this time, the data collection will be stopped and the data transmission can be carried out. The service node will send the collected data to the cluster head for simple processing. When the service node transmits all its perceived data, it transitions to a sleep state, stops sensing and transmitting tasks, and reduces energy consumption. See Fig. 5. Under the same perceptive task, when the task for which the service node is responsible is completed, it will change from a working state to a sleeping state, and repeat the transition process of the above state until the new perceptive task starts.

Cluster group member selection

In traditional clustering protocol, the size of the cluster range is determined by the communication range of the cluster-head node. Due to the randomness of node distribution and the inaccessibility of its sensing range, the traditional clustering method cannot achieve a good effect in the mobile swarm intelligent perception network. Considering the characteristics of the MCS such as large scale, strong mobility, and long transmission distance, there are higher requirements for selecting cluster member nodes when dividing clusters. Therefore, a method is proposed to specify cluster radius for the structure of the group perceptive network, R is the cluster radius for the node under consideration, which determines the communication scope within the mobile crowd-sensing network. As shown in formula (1):

$$R = \rho(N_i)\left(1 - c\frac{d_{\max} - d_{ns}}{d_{\max} - d_{\min}}\right)R_0 \tag{1}$$

where, $\rho(N_i)$ is the density of nodes within one hop of node i, reflecting the local node population density which influences the cluster radius. $d_{\max}$ is the maximum distance observed between any perceived node and the cloud server across the entire network, representing the farthest node reach. $d_{\min}$ is the minimum distance observed between any perceived node and the cloud server across the entire network, indi-

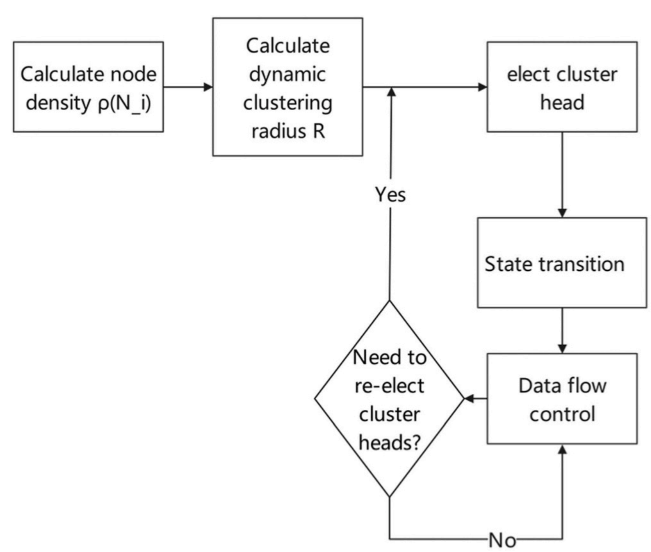

Fig. 4. Flowchart of FITNESS AND NODE DENSITY CLUSTERING MECHANISM IN MCS.

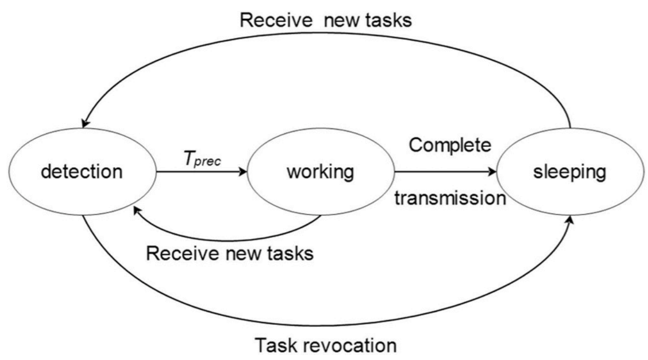

Fig. 5. Node state transition diagram.

cating the nearest node position. And d_{ns} is the actual distance between node i and the cloud server, used to adjust the influence of node position on cluster sizing. The communication radius is jointly determined by $\rho(N_i)$, c, and R_0. c is the parameter that modifies the impact of the difference between $d_{\max}$ and d_{ns}. R_0 is predefined maximum communication range of a network node, serving as a baseline for the maximum potential cluster radius.

The cluster radius R is dynamically adjusted based on the relative position of node i to the cloud server, modulated by the density of nearby nodes. This mechanism ensures that nodes farther from the cloud server can have a larger cluster radius, facilitating efficient data aggregation and transmission within areas of higher node density. Conversely, nodes closer to the server have a smaller cluster radius, optimizing network resources and reducing congestion. This adaptive mechanism is crucial for managing the challenges of large-scale, highly mobile, and long-distance transmissions in crowd-sensing networks.

The cluster head is responsible for the sensing work in each sensing area. The cluster head election method is described in the following research. The divided perception network is shown in Fig. 6, and its clustering method is centered on the cluster head. The topology of the entire cluster changes with the movement of the cluster head node and the change of the cluster head radius.

Cluster head election method

The selection of cluster heads is crucial for the stability and longevity of the entire network. Traditional methods of selecting cluster heads often involve a degree of randomness, leading to significant fluctuations in the number of cluster heads during elections and failing to consider the coverage features crucial for Mobile Crowd-Sensing opportunities. Merely incorporating a conventional clustering algorithm into the node perception phase of the Mobile Crowd-Sensing network structure does not systematically or standardly process perception information. Instead, it increases network energy consumption and shortens the network's life cycle.

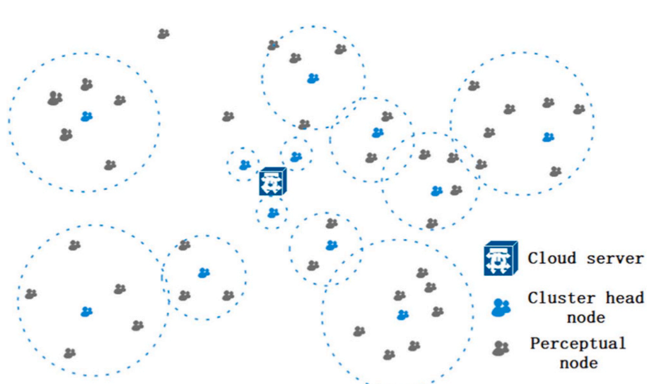

Fig. 6. Crowd sensing area division.

To enhance the network's self-adaptability, the election of cluster heads should not solely focus on energy consumption but also consider speed and movement direction. Both the sensing nodes and the elected cluster heads are mobile, displaying relative speeds during the election process.

The following issues can arise with the use of random cluster head elections:

- If a common sensing node at the edge of the cluster is elected as the cluster head, it may move out of the cluster, necessitating a re-election for the entire cluster and resulting in excessive energy consumption.
- If a node moves too rapidly, it will remain within the cluster for only a short time after being elected as the cluster head, which can shorten the overall life cycle of the cluster and negatively impact subsequent data fusion and forwarding.
- Electing a sensing node with low energy as the cluster head can lead to its rapid depletion and premature death of the cluster.

Based on an analysis of these issues and considering the movement patterns of the sensed nodes and cluster heads, the movement directions of the four nodes as depicted in Fig. 7 were determined.

The primary focus of this algorithm is to optimize the selection of cluster heads by considering both the residual energy of the nodes and their projected duration within the cluster. By selecting a node that not only possesses the highest energy fitness but also demonstrates prolonged stability within the cluster, the algorithm enhances the longevity and coordination capabilities of the network's nodes. This approach helps prevent the collapse of the cluster due to the premature depletion of the cluster head's energy, thus significantly extending the network's overall lifecycle, as illustrated in formula (2). This strategy ensures that the chosen cluster head can maintain its role for an extended period, thereby stabilizing the network's structure and functionality.

$$f = \omega_1 \times E + \omega_2\left(\frac{\sqrt{X^2 + R^2 - 2XR\cos\beta}}{F(V)}\right) \tag{2}$$

$$E = E_{init} - n\left(E_{elect} + E_{fs}d^2 + E_{mp}d^4\right) - mE_{dafu} \tag{3}$$

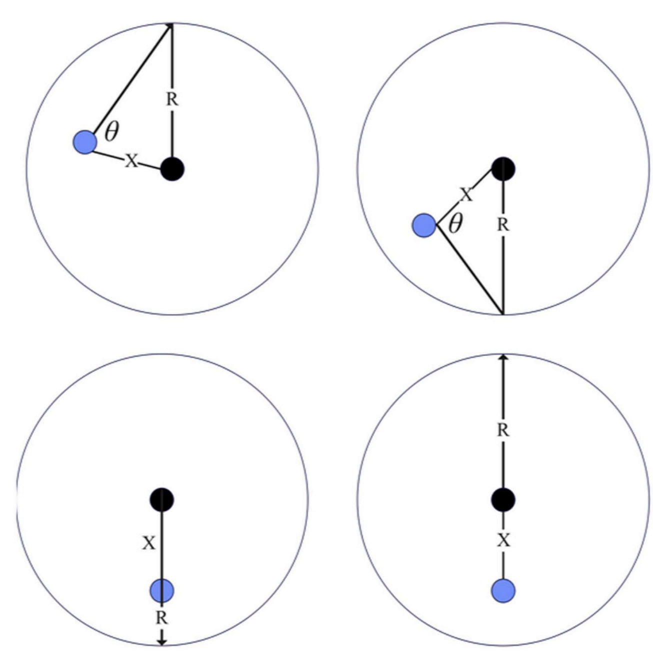

Fig. 7. Node movement direction.

$$X = \sqrt{(x_i - x_c)^2 - (y_i - y_c)^2} \tag{4}$$

$$\beta = \cos^{-1}\left(\frac{(x_i - x_c)x_c + (y_i - y_c)y_c}{\sqrt{(x_i - x_c)^2 - (y_i - y_c)^2}\sqrt{x_c^2 + y_c^2}}\right) \tag{5}$$

$$F(V) = \sqrt{(v_{ix} - v_{hx})^2 + (v_{iy} - v_{hy})^2} \tag{6}$$

Where, E_{init} is Initial energy of the node, n is the total bits transmitted by the node. d is transmission distance in meters (m). m is the number of data aggregation operations performed, E_{elect} is energy per bit for electronic processing, E_{fs} is the energy per bit per square meter for free-space transmission, E_{mp} is the energy per bit per cubic meter for multipath data transmission, E_{dafu} is the energy per data aggregation operation. X is the distance to the previous cluster head, x_i, y_i are the coordinates of the node, x_c, y_c are the coordinates of the previous cluster head. β is the angle between X and R. $F(V)$ is the relative velocity function, v_{ix}, v_{iy} are the velocity components of the node, and v_{hx}, v_{hy} are the velocity components of the cluster head. ω_1, ω_2 are weighting factors for energy and distance-velocity terms, $\omega_1 = 0.6, \omega_2 = 0.4$. This mathematical model provides a framework for evaluating the suitability of nodes as cluster heads by considering not only their energy capacity but also their position and mobility relative to the network structure.

Fig. 8 illustrates the cluster head election process. In the described mobile sensing network, sensing nodes are randomly distributed across

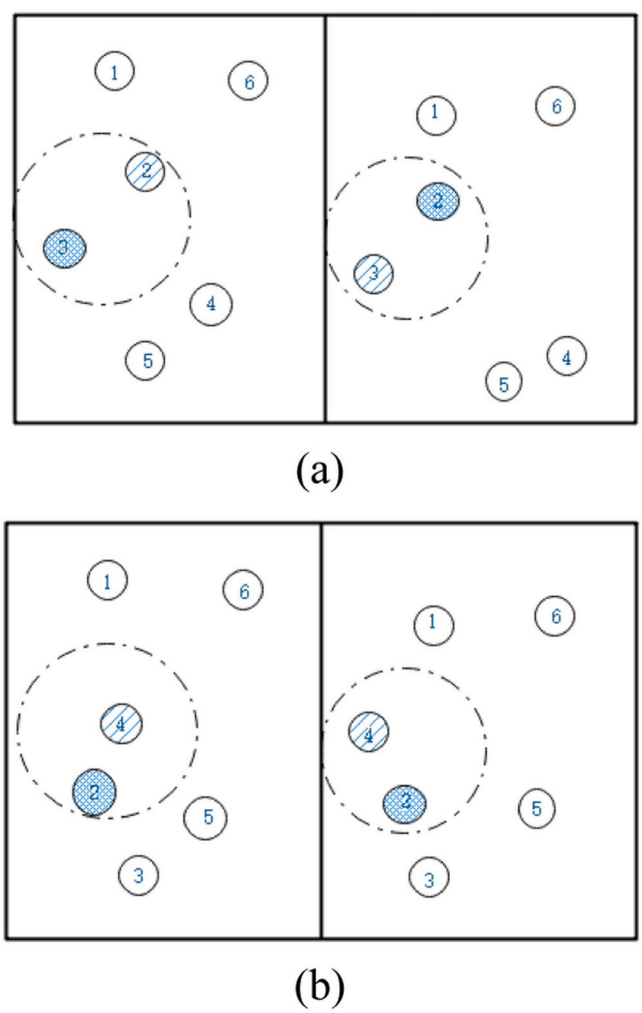

Fig. 8. The fitness is optimal at t1. (a) The fitness is optimal at t2. (b) Optimal fitness at time t_2.

the network topology, analogous to stars within a black hole optimization algorithm framework. Each node (or "star") is assigned an adaptation value, denoted by f. This value quantifies each node's suitability to act as a cluster head, based on factors like energy, proximity, and stability.

$$CH = \begin{cases} Node_i, \mathit{if}\ f_i > f_j\ \mathit{for\ all}\ j \neq i \\ Node_j, \mathit{if}\ f_i \leq f_j\ \mathit{and\ no\ other}\ f_k > f_j \end{cases} \tag{7}$$

Where, $Node_i$ is the any node in the network where i is its index., $Node_j$ is the previous cluster head, where j represents its index. The process involves evaluating whether any node

i in the network has an adaptation value f_i that surpasses that of the previous cluster head f_j. If such a node exists, it is selected as the new cluster head. Otherwise, the previous cluster head j retains its position. This method ensures that the new cluster head is at least as capable as the previous one, if not more so, thereby potentially enhancing network stability and efficiency.

This refined model and description clearly define how the transition of leadership is managed within the network, ensuring continuity and stability in cluster head roles.

Algorithm description

Here is the refined and corrected pseudo-code for the clustering algorithm used in mobile group perception. This algorithm involves selecting cluster heads based on their fitness values for optimization and broadcasting this information to cluster members.

Algorithm 1. Mobile crowd-sensing network clustering algorithm.

```
Nodes: List of all sensing nodes in the network.
R_0: Predefined maximum communication range of a node.
Density ρ(N_i): Density of nodes within one hop of node i.
d_max, d_min Maximum and minimum distances between any node
and the cloud server across the network.
d_{ns}: Distance between node i and the cloud server.
Clusters: Array to store clusters with each cluster having a list of nodes
and a cluster head.

Clusters = []
For each node i in Nodes:
    Initialize a new cluster C_i with node i as the temporary
    cluster head
    Clusters.append(C_i)
For each node i in Nodes:
    Calculate cluster radius R_i for node i:
    R_i = ρ(N_i) * (1 - c * (d_max - d_{ns}) / (d_max - d_min)) *
    R_0
    For each node j in Nodes:
        If distance between node i and j <= R_i:
            Add node j to cluster C_i if not already in a cluster
For each cluster C_i in Clusters:
    best_fitness = -infinity
    For each node k in C_i:
        Calculate fitness f_k:
        f_k = ω_1 × E_k + ω_2 × (sqrt((x_k - x_c)^2 + (y_k -
        y_c)^2 - 2(x_k - x_c)(y_k - y_c)cos(β_k))/F(V_k))
        If f_k > best_fitness:
            best_fitness = f_k
            C_i.head = k  # Assign new cluster head
Periodically or upon significant node movement:
    For each cluster C_i:
        Re-evaluate cluster memberships based on new node
        positions
        Re-run Cluster Head Election if necessary
For each cluster C_i:
    Collect data from members of C_i
    Aggregate data at C_i.head
    Transmit aggregated data to cloud server or next higher layer
```

Table 1
The simulation parameters for the traditional clustering.

Parameter name	Parameter value	Explanation
Simulation Area	200 m × 200 m	The physical area covered by the simulation.
Number of Nodes	200	The total number of nodes in the network.
E_{init}	0.5 J	Initial energy of nodes.
E_{elect}	50 nJ/bit	Energy consumed for data transmission per bit.
E_{fs}	50 nJ/bit/m^2	Energy consumed for free-space data transmission per bit per square meter.
E_{mp}	0.0013 pJ/bit/m^4	Energy consumed for multi-path data transmission per bit per cubic meter.
E_{dafu}	5 nJ	Energy consumed for data aggregation.

Simulation result

The simulation scene

This study introduces an advanced clustering algorithm designed for mobile crowd-sensing networks. It highlights the dynamic adjustment of cluster radii based on node density and proximity to cloud servers. The algorithm also enhances the election process for cluster heads by considering node stability and energy levels, aiming to extend the network's lifecycle and improve resource management.

The feasibility of the algorithm is demonstrated through simulations that compare it with established traditional clustering protocols such as LEACH, LEACH-C, SEP, DEEC, and the mobile-specific LEACH-M. These simulations evaluate various metrics, including energy consumption, node longevity, data transmission, and network balancing. The simulation parameters are detailed in Table 1.

The simulation experiments primarily focus on the impact of various factors, including node density, speed, distance from the cluster head, energy, and distance from the base station on network lifespan and node energy consumption. The comparison between the proposed algorithm and traditional clustering algorithms is conducted through these simulation experiments.

Comparison of classic clustering algorithms

Lifetime experiment

This section presents a comparative analysis of network lifecycle experiment results. The lifecycle of a network is a critical factor that influences its performance. Fig. 9 illustrates the comparison between the MCLEC algorithm and a classical clustering algorithm in terms of node mortality over the same period. A node is defined as "dead" when its energy is depleted, indicated by a value of 0, rendering it inactive.

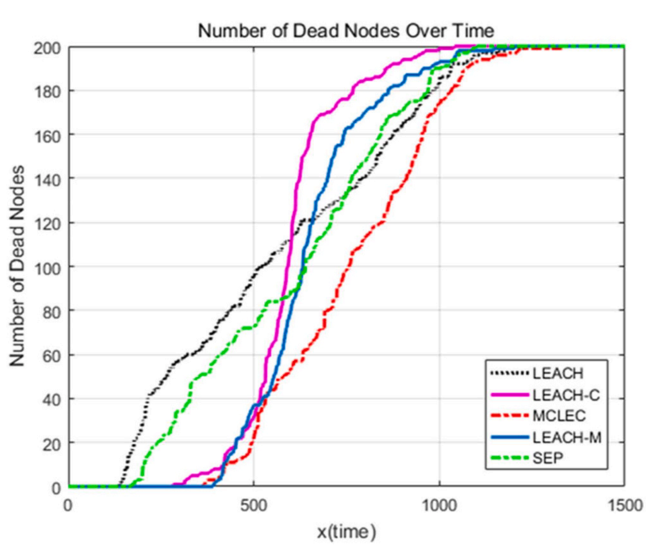

Fig. 9. Comparison of the number of death nodes.

In the initial phase of the mobile group perceptron network, it is essential to cluster nodes based on their energy levels, while node mobility remains uncontrollable. Consequently, all nodes must broadcast to ascertain the number of nodes within their communication radius. Over time, the mortality rate for nodes in all routing algorithms doubles due to the substantial energy consumed in communication, data transmission, and cluster head selection.

After x = 500, the MCLEC algorithm demonstrates a lower rate of node deaths compared to the classical clustering protocol at the same point in time. This improvement is due to the algorithm's cluster head election mechanism, which considers various factors such as node energy and movement speed, significantly enhancing node survival compared to the LEACH-M algorithm for mobile sensors. However, as runtime exceeds 1000, the growth curve of the MCLEC algorithm steepens, indicating an accelerated rate of node mortality.

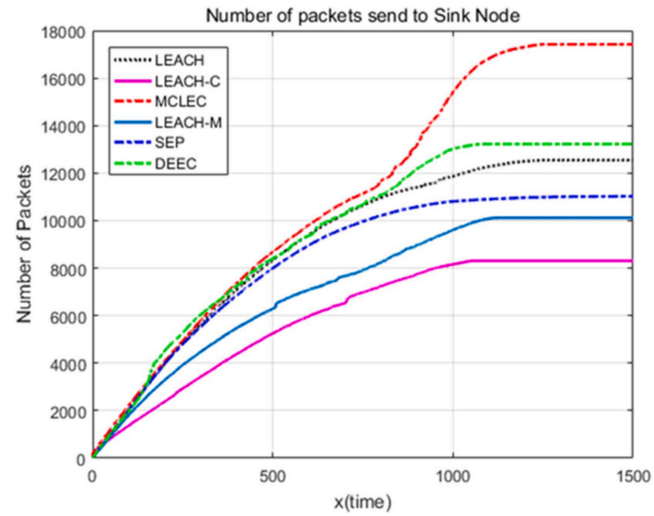

Fig. 10. Packet to sink node comparison chart.

Fig. 10 displays a contrast diagram showing the number of packets nodes send to the gathering node. As illustrated, while packet counts in other networks have ceased to grow, those in the MCLEC network continue to increase. This indicates that the MCLEC network has a longer lifecycle and sends packets over a longer duration compared to other algorithms, thus transmitting a greater number of packets to the gathering node than other clustering algorithms.

Fig. 11 presents a comparison of energy consumption among different clustering algorithms. Although MCLEC initially consumes energy more rapidly, it proves more efficient in terms of energy consumption during the middle and late stages of its operation. This improvement is attributed to the MCLEC algorithm's ability to shorten data transmission distances, maintain clustering ranges within reasonable limits, and minimize the extra energy consumption typically caused by large-scale broadcasting. Overall, the energy efficiency of the MCLEC is superior to that of other clustering algorithms.

In Fig. 12 illustrates three scenarios: the time at which the first node dies, when 10 % of the nodes are dead, and when all nodes die. The experimental data suggest that MCLEC has a longer lifecycle than other traditional clustering algorithms. The time of the first node's death is critical as it sets the precedent for network stability. Subsequent node deaths are spread over time, ultimately defining the lifecycle of the entire network. The experimental results indicate that MCLEC not only enhances network stability but also extends the overall lifecycle compared to other classic clustering algorithms.

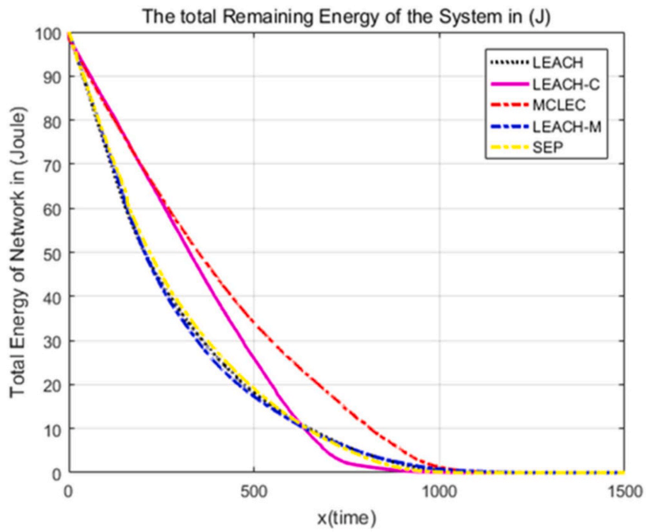

Fig. 11. Energy consumption comparison chart.

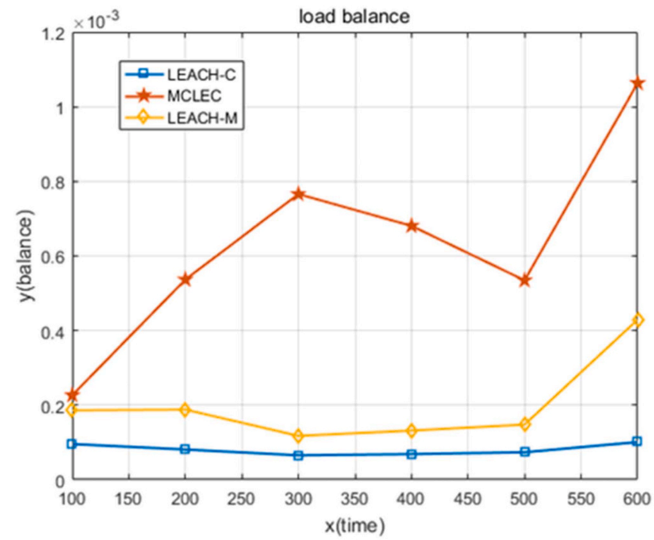

Fig. 13. LBF comparison.

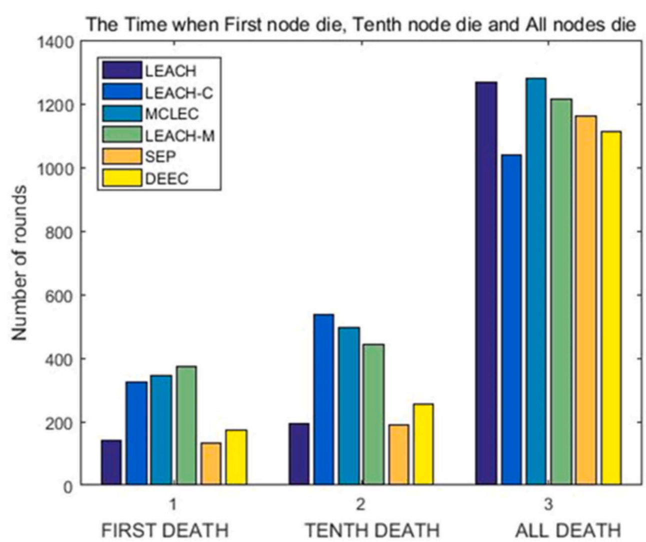

Fig. 12. Node life cycle comparison chart.

Transmission experiment

The number of nodes within each cluster significantly influences the network's clustering structure and is crucial for assessing whether the structure is reasonable. The node count impacts both the real-time performance of data transmission and the energy consumption of the nodes. Whether employing the "storage-carriage-forwarding" communication mode in this algorithm or the "aggregation-forwarding" mode used in traditional clustering algorithms, a one-hop transmission mode between nodes and cluster heads is utilized. Key performance indicators include time delay, load balancing, and routing overhead.

In the clustering algorithm, the number of nodes per cluster affects the network's survival time and transmission efficiency. The degree of load balancing within a cluster is a key metric for evaluating a clustering algorithm. Thus, a Load Balancing Factor (LBF) is introduced to assess the performance of the clustering algorithm. The formula for LBF is given by::

$$LBF = \frac{CH\_num}{\sum_{i=1}^{CH\_num} (x_i - \overline{x})^2}$$

where, *CH_num* is the number of cluster heads in the network, x_i is the number of members contained in the ith cluster, and x is the average number of members of the cluster. Fig. 13 compares the load balancing experiments of LEACH, LEACH-C, LEACH-M, and MCLEC. A higher LBF value indicates a more balanced load within the cluster, signifying more reasonable clustering. According to the experimental data, the MCLEC algorithm shows a significantly higher load balancing value than the other algorithms, indicating it is more efficient in energy conservation and effectively extends the lifecycle.

As depicted in Fig. 14, a comparison of network transmission delays reveals that MCLEC experiences relatively large fluctuations in transmission delay. When the node count is low, there is a noticeable increase in transmission delay. Experimental data show that MCLEC's transmission delay is notably better than that of LEACH and LEACH-M, and it is gradually approaching the performance of the LEACH-C algorithm. MCLEC adopts an "edge" concept, where the edge is not only a geographic boundary but also involves priority processing of data before packets reach the cloud, contributing to the variation in delay.

Routing overhead is a vital metric for evaluating the routing performance of a network. Experimental data demonstrates changes in routing costs as the number of nodes varies from 100 to 220. As shown in Fig. 15, routing overhead increases with the number of nodes due to higher network density, more intense competition for channel resources, and more packet copies forwarded by nodes in the network. The data indicate that MCLEC can effectively control routing overhead compared to the other three algorithms, with significantly lower overhead after the node count exceeds 140.

Comparison of advanced and optimized WSN clustering techniques

In addition to traditional classical clustering algorithms, advanced and optimized WSN clustering algorithms have recently become a critical area of research due to their significant impact on network lifetime and energy efficiency. The figures presented in this study offer a comparative analysis of several advanced and optimized WSN clustering algorithms, specifically focusing on their residual energy percentages over time and overall network lifetimes under various simulation conditions. The algorithms analyzed in this study include E-FLZSEPFCH [34], DFLC [35], ECPF [36], ACAWT [34], UCR [37], CHEF [38], and Gupta's algorithm [39]. The simulation parameters for the advanced and optimized clustering are shown in Table 2.

Fig. 16 shows the residual energy percentages at various simulation times for a range of novel clustering algorithms, including MCLEC, E-FLZSEPFCH, DFLC, ECPF, ACAWT, UCR, CHEF, and Gupta. The analysis

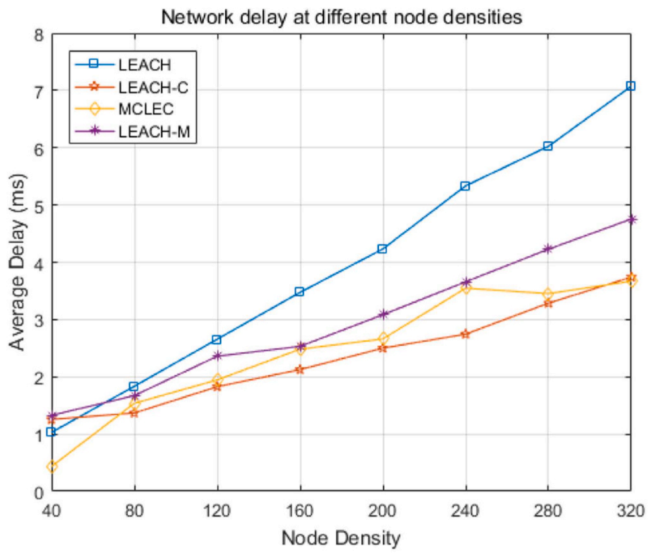

Fig. 14. Average delay comparison.

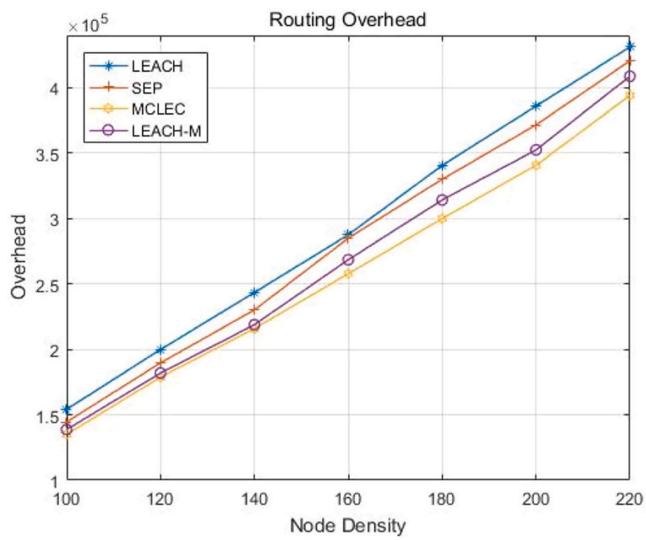

Fig. 15. Routing overhead comparison.

Table 2
The simulation parameters for the advanced and optimized clustering.

Parameter name	Parameter value	Explanation
Simulation Area	1050 × 1100	The physical area covered by the simulation.
Number of Nodes	1550	The total number of nodes in the network.
E_{init}	3 J	Initial energy of nodes.
E_{elect}	50 nJ/bit	Energy consumed for data transmission per bit.
E_{fs}	10 pJ/bit/m^2	Energy consumed for free-space data transmission per bit per square meter.
E_{mp}	0.0013 pJ/bit/m^4	Energy consumed for multi-path data transmission per bit per cubic meter.
E_{dafu}	5 nJ	Energy consumed for data aggregation.
Data packet size (bytes)	150 bytes	Bytes forming a data packet for transmission.
Control packet size (bytes)	35 bytes	Bytes in packet for network control information.

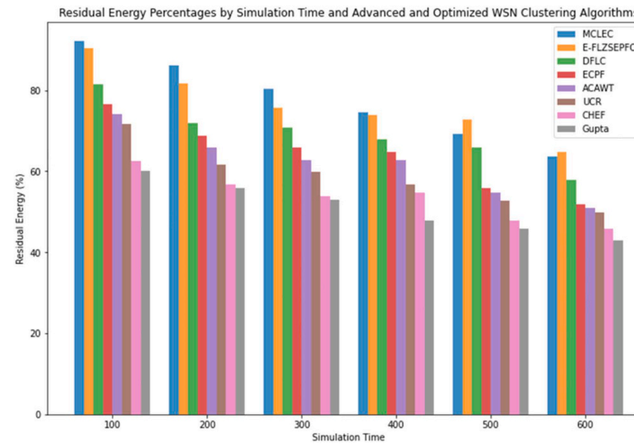

Fig. 16. Residual energy percentages by simulation time and advanced and optimized WSN clustering algorithms.

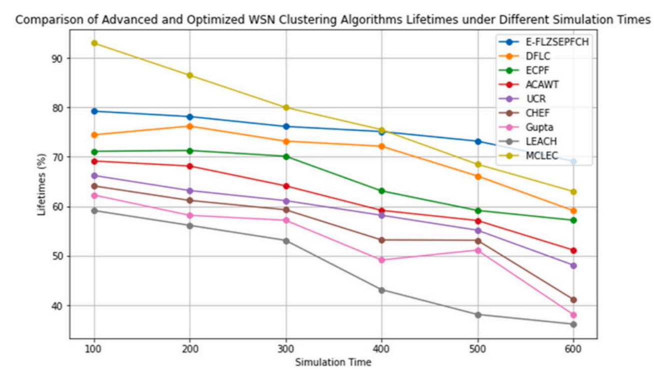

Fig. 17. Comparison of advanced and optimized WSN clustering algorithms lifetimes under different simulation times.

reveals that MCLEC consistently demonstrates the best energy retention across all time points, with a relatively flat energy consumption curve, indicating its excellent energy management capabilities. The E-FLZSEPFCH algorithm performs particularly well in the mid-phase, with slow energy decay, suggesting effective energy efficiency strategies during this period. In contrast, other algorithms such as DFLC, ECPF, ACAWT, UCR, CHEF, and Gupta show more significant energy declines in later stages, especially Gupta and CHEF, which exhibit weaker energy retention capabilities. These data highlight the importance of considering energy efficiency when selecting clustering algorithms.

Fig. 17 compares the lifetime metrics of the MECLEC algorithm with other novel clustering algorithms. In the performance comparison between MECLEC and E-FLZSEPFCH, although E-FLZSEPFCH shows better stability and slower performance degradation after 400-time units, overall, the MECLEC algorithm still outperforms E-FLZSEPFCH as well as other algorithms throughout all the simulated time periods. The graph emphasizes the potential advantages of the MECLEC algorithm when designing WSN systems for long-term deployment. Despite the durability shown by E-FLZSEPFCH in the later stages, the superior overall performance of MECLEC indicates that it can maintain a higher performance level over the long run. Particularly in the early stages of the network, MECLEC's high efficiency can significantly extend the effective operational time of the network, which is especially important for applications that require high efficiency from the start.

Conclusion

This study introduces a novel clustering methodology to mobile crowd-sensing networks, achieving balanced energy consumption among the sensing nodes. By precisely defining the cluster radius and thoroughly evaluating various determinants that influence cluster head selection, a new clustering algorithm predicated on adaptability and node density has been formulated. This algorithm substantially extends the network's lifecycle and fortifies its stability. Comparative analysis with leading-edge clustering algorithms indicates that while this approach significantly diminishes energy consumption and ameliorates transmission delays relative to conventional clustering networks, it also exhibits inherent limitations that necessitate further enhancements for robust, large-scale deployment. Furthermore, this research presupposes an intrinsic trust among nodes, thereby omitting an analysis of privacy and security concerns. Future endeavors will focus on scaling the network, augmenting its durability, and prioritizing the security and privacy of users.

Abbreviations

MCS: Mobile Crowd Sensing
IoT: Internet of Things
MCLEC: Mobile Crowd-sensing Low Energy Clustering Algorithm
LEACH: Low-Energy Adaptive Clustering Hierarchy
LEACH-C: Low-Energy Adaptive Clustering Hierarchy Centralized
LEACH-M: Low-Energy Adaptive Clustering Hierarchy for mobile
LBF: Load-Balanced Fuzzy
DEEC: Distributed Energy-Efficient Clustering
SEP: Stable Election Protocol
R_0: Predefined maximum communication range of a node
$\rho(N_i)$: Density of nodes within one hop of node i
d_{max}: Maximum distance between any node and the cloud server across the network
d_{min}: Minimum distance observed between any perceived node and the cloud server across the entire network
d_{ns}: Actual distance between node i and the cloud server
c: Adjustment parameter affecting the influence of distance on cluster radius
n: Total bits transmitted by the node
d: Transmission distance in meters
m: Number of data aggregation operations performed
E: Residual energy of a node
E_{init}: Initial energy of the node
E_{fs}: Energy per bit per square meter for free-space transmission
E_{elect}: Energy consumed for data transmission per bit
E_{mp}: Energy per bit per cubic meter for multipath data transmission.
E_{dafu}: Energy per data aggregation operation
x_i, y_i: Coordinates of the node
x_c, y_c: Coordinates of the previous cluster head
X: Distance to the previous cluster head
R: Cluster radius
β: Angle between the position vector to the previous cluster head and the relative movement direction
F(V): Relative velocity function
v_i, v_i: Velocity components of the node
v_{hx}, v_{hy}: Velocity components of the cluster head
ω_1, ω_2: Weighting factors for energy and distance-velocity terms.

CRediT authorship contribution statement

Tonghui Qu: Data curation, Investigation. **Kaiyu Wang:** Formal analysis, Investigation, Software. **Hongwei Wang:** Conceptualization, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Hao Li:** Data curation, Formal analysis, Writing – original draft, Writing – review & editing. **Jun Ma:** Data curation, Formal analysis, Project administration. **Xunhuan Ren:** Investigation, Software.

Declaration of Competing Interest

All authors declare there are no competing interest of publishing this manuscript.

References

[1] R.K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, IEEE Commun. Mag. 49 (11) (2011) 32–39.
[2] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, et al., Crowdsourcing with smartphones, IEEE Internet Comput. 16 (5) (2012) 36–44.
[3] H. Ma, D. Zhao, P. Yuan, Opportunities in mobile crowd sensing, IEEE Commun. Mag. 52 (8) (2014) 29–35.
[4] J. Weppner, P. Lukowicz, Bluetooth based collaborative crowd density estimation with mobile phones, in: Proceedings of the 2013 IEEE International conference on pervasive computing and communications (PerCom), IEEE, 2013, pp. 193–200.
[5] E. Koukoumidis, L.S. Peh, M.R. Martonosi, SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, ACM, 2011, pp. 127–40.
[6] P. Simoens, Y. Xiao, P. Pillai, , Scalable crowd-sourcing of video from mobile devices, in: Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2013, pp. 139–52.
[7] S.B. Eisenman, E. Miluzzo, N.D. Lane, et al., BikeNet: a mobile sensing system for cyclist experience mapping, ACM Trans. Sens. Netw. (TOSN) 6 (1) (2009) 6.
[8] J. Eriksson, L. Girod, B. Hull, , The pothole patrol: using a mobile sensor network for road surface monitoring, in: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, ACM, 2008, pp. 29–39.
[9] Z. Yang, C. Wu, Y. Liu, Locating in fingerprint space: wireless indoor localization with little human intervention, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, ACM, 2012, pp. 269–80.
[10] A. Rai, K.K. Chintalapudi, V.N. Padmanabhan, et al., Zee: zero-effort crowdsourcing for indoor localization, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, ACM, 2012, pp. 293–304.
[11] H. Li, X. He, S. Ding, Routing algorithm for reducing packet loss in mobile WSN, in: Proceedings of the 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2019, pp. 258–63.
[12] R.V. Kulkarni, A. Forster, G.K. Venayagamoorthy, Computational intelligence in wireless sensor networks: a survey, IEEE Commun. Surv. Tutor. 13 (1) (2010) 68–96.
[13] L. Liu, X. Qi, J. Xue, et al., A topology construct and control model with small-world and scale-free concepts for heterogeneous sensor networks, Int. J. Distrib. Sens. Netw. 10 (3) (2014) 374251.
[14] M.Y. Khan, N. Javaid, M.A. Khan, et al., Hybrid DEEC: towards efficient energy utilization in wireless sensor networks, arXiv preprint arXiv:1303.4679, 2013.
[15] A.T. Campbell, S.B. Eisenman, N.D. Lane, et al., People-centric urban sensing, in: Proceedings of the 2nd Annual International Workshop on Wireless Internet, ACM, 2006, 18.
[16] S.B. Eisenman, A. Campbell, People-centric Mobile Sensing Networks, Columbia University, 2008.
[17] D. Zhao, H. Ma, S. Tang, et al., COUPON: A cooperative framework for building sensing maps in mobile opportunistic networks, IEEE Trans. Parallel Distrib. Syst. 26 (2) (2014) 392–402.
[18] L. Wang, D. Zhang, Z. Yan, et al., effSense: a novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading, IEEE Trans. Syst. Man Cybern.: Syst. 45 (12) (2015) 1549–1563.
[19] F. Xiao, Z. Jiang, X. Xie, et al., An energy-efficient data transmission protocol for mobile crowd sensing, Peer-to-Peer Netw. Appl. 10 (3) (2017) 510–518.
[20] G. Yang, C. Li, H. Yang, et al., Energy-efficient communication protocol for wireless microsensor networks, Inf. Technol. J. 12 (12) (2009) 537–568.
[21] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan,Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE, vol. 2, 2000, 10 pp.
[22] L.T. Nguyen, X. Defago, R. Beuran, , An energy efficient routing scheme for mobile wireless sensor networks, in: Proceedings of the 2008 IEEE International Symposium on Wireless Communication Systems, IEEE, 2008, pp. 568–72.
[23] L. Qing, Q. Zhu, M. Wang, Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks, Comput. Commun. 29 (12) (2006) 2230–2237.
[24] S. Panbude, B. Iyer, A.B. Nandgaonkar, et al., DFPC: dynamic fuzzy-based primary user aware clustering for cognitive radio wireless sensor networks, Eng., Technol. Appl. Sci. Res. 13 (6) (2023) 12058–12067.
[25] R.B. Pedditi, K. Debasis, MACR: a novel meta-heuristic approach to optimize clustering and routing in IoT-based WSN, Int. J. Intell. Syst. Appl. Eng. 12 (1) (2024) 346–359.
[26] P. Rajpoot, P. Dwivedi, MADM based optimal nodes deployment for WSN with optimal coverage and connectivity, in: IOP Conference Series: Materials Science and Engineering, 1020, IOP Publishing, 2021 012003.
[27] P. Rajpoot, P. Dwivedi, Optimized and load balanced clustering for wireless sensor networks to increase the lifetime of WSN using MADM approaches, Wirel. Netw. 26 (1) (2020) 215–251.

[28] S.H. Singh, K.R. Verma, P. Rajpoot, Partition based strategic node placement and efficient communication method for WSN, in: Proceedings of the 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, 2018, pp. 1807–12.

[29] T.K. Dao, An optimal node localization in WSN based on siege whale optimization algorithm, Cmes-Comput. Model. Eng. Sci. 138 (3) (2024).

[30] P. Chithaluru, F. Al-Turjman, T. Stephan, et al., Energy-efficient blockchain implementation for cognitive wireless communication networks (CWCNs), Energy Rep. 7 (2021) 8277–8286.

[31] K. Dubey, P. Rajpoot, A.K. Singh, , Fuzzy based technique for nodes coverage with load balancing data collection using multiple conflicting factors, in: Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), IEEE, 2019, pp. 579–84.

[32] P. Rajpoot, P. Dwivedi, Matrix method for non-dominated sorting and population selection for next generation in multi-objective problem solution, in: Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2018, pp. 670–6.

[33] S. Tumula, Y. Ramadevi, E. Padmalatha, et al., An opportunistic energy-efficient dynamic self-configuration clustering algorithm in WSN-based IoT networks, Int. J. Commun. Syst. 37 (1) (2024) e5633.

[34] A. Ali, A. Ali, F. Masud, et al., Enhanced fuzzy logic zone stable election protocol for cluster head election (E-FLZSEPFCH) and multipath routing in wireless sensor networks, Ain Shams Eng. J. 15 (2) (2024) 102356.

[35] A. Alaybeyoglu, A distributed fuzzy logic-based root selection algorithm for wireless sensor networks, Comput. Electr. Eng. 41 (2015) 216–225.

[36] H. Taheri, P. Neamatollahi, O.M. Younis, et al., An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic, Ad Hoc Netw. 10 (7) (2012) 1469–1481.

[37] G. Chen, C. Li, M. Ye, et al., An unequal cluster-based routing protocol in wireless sensor networks, Wirel. Netw. 15 (2009) 193–207.

[38] J.M. Kim, S.H. Park, Y.J. Han, , CHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks, in: Proceedings of the 2008 10th International Conference on Advanced Communication Technology, IEEE, vol. 1, 2008, pp. 654–9.

[39] P. Gupta, S. Tripathi, S. Singh, Energy-efficient routing protocols for cluster-based heterogeneous wireless sensor network (HetWSN)—strategies and challenges: a review, Data Anal. Manag.: Proc. ICDAM (2021) 853–878.