A Cooperative Block-variant Monitoring Mechanism Based on Spectral Clustering for Internet of Things

Jiaxi Chen¹,a, Xingchuan Liu¹,b, Jinyun Wu¹,c, Zhiqiang Ding²,d, Yue Cui³,e and Shuo Yang⁴,f

¹The Smart City Research Institute of China Electronics Technology Group Corporation, Shenzhen, China
²Shanxi Maintenance Company of SGCC, Shanxi, China
³Xi’an Jiaotong University, Xi’an, China

¹1021543242@qq.com;²liuxingchuan@ctec.com.cn;³wujinyun@ctec.com.cn;⁴774556514@qq.com;⁵N924925465@stu.xjtu.edu.cn;⁶suiningys@126.com

Abstract. There exist two main defects in traditional monitoring systems: rigid monitoring intensities and fixed task roles, which induce low monitoring efficiencies and large power consumptions. To alleviate these two problems, this paper proposes a cooperative block-variant monitoring mechanism for Internet of Things. This mechanism divides the whole monitoring area into several blocks with different monitoring intensities according to the spatial distribution of monitoring terminals based on spectral clustering. The monitoring intensities, including monitoring densities and frequencies, are decided by the status of monitoring objects in different blocks and the values of pre-setting thresholds. By adjusting the monitoring densities and frequencies in real time, this method makes the monitoring focus on the most critical blocks, which improves the monitoring efficiencies and reduces the overall consumption of system. In addition, adequate switching of the task roles of nodes balances their workloads, and therefore extends the overall life of the monitoring systems. A large number of experiments have been carried out, and the results show that this collaborative monitoring mechanism achieves good performance.

1. Introduction
Multi-terminal cooperative monitoring is one of the common scenarios in the field of Internet of things [1]. Traditional multi-terminal monitoring systems face challenges of heavy maintenances and power consumptions due to the lack of terminal cooperation capability and pertinence in monitoring [2] [3]. In order to improve the monitoring efficiency of terminal system, the idea of this paper is to divide the monitoring area into several blocks according to the spatial distribution of the terminals, using the spectral clustering method [4]. After the environmental variables are detected in all monitoring blocks, the core node of each block computes its monitoring densities and frequencies currently needed [5] [6]. Then each block will randomly select monitoring nodes and communication nodes according to the required monitoring densities, while unnecessary monitoring nodes will turn on sleep mode until the next time period, which helps to reduce the power consumption of overall system.

In this paper, the similarity graph is introduced to represent the topology of multi-terminal monitoring systems [7]. Then, the principle and steps of spectral clustering are applied to solve the problem of block division. Besides, the settings of thresholds are listed, paving the way of adjusting
monitoring status [8]. After that, the method of determining the monitoring intensities, including monitoring densities and frequencies, are detailed described. Finally, the expected and experimental monitoring effects are shown in detail, to test the effectiveness of the proposed monitoring mechanism.

2. Methods

The proposed mechanism utilizes a series of methods to enhance the monitoring efficiencies and reduce power consumption of multi-terminal monitoring systems, including similarity graph representation, spectral clustering, thresholds pre-setting, and monitoring intensity [9]. The use of these methods eventually leads to the expected and experimental monitoring results.

2.1. Similarity Graph Representation

Given the positions of \( N \) nodes \( \{x_1, x_2, \cdots, x_N\} \) and some notion of similarity \( s_{ij} \geq 0 \) between all pairs of nodes of multi-terminal monitoring systems, it is easy to represent the data in form of similarity graph \( G=(V, E) \), where vertex \( v_i \in V \) represents a data point \( x_i \), and two vertices \( v_i \) and \( v_j \) are connected if the similarity \( s_{ij} \geq \epsilon \), where \( \epsilon \) is a given positive constant[10]. The similarities of all pairs of nodes forms the similarity matrix \( S=[s_{ij}]_{N \times N} \), which is the basis of spectral clustering.

2.2. Spectral Clustering

Spectral clustering aims at separating the similarity graph \( G=(V, E) \) into several blocks, so as to implement the process of block-variant monitoring [11]. The main steps of spectral clustering are shown as follows:

Step 1: Input the similarity matrix \( S \) and clustering number \( K \).
Step 2: Compute the degree matrix \( D \):

\[
D = \text{diag}\{d_1, d_2, \cdots, d_N\}
\]

Where \( d_i = \sum_{j=1}^{N} s_{ij} \), and index \( i = 1, 2, \cdots, N \).

Step 3: Compute the Laplacian matrix \( L \):

\[
L = D - S
\]

Step 4: Normalize the Laplacian matrix to get the symmetric Laplacian matrix \( L_{sym} \):

\[
L_{sym} = \frac{1}{\sqrt{D}} L \frac{1}{\sqrt{D}}
\]

Step 5: Compute eigenvalues and eigenvectors of symmetric Laplacian matrix \( L_{sym} \):

\[
L_{sym} u_i = \lambda_i u_i
\]

Where \( \lambda_i \) is the eigenvalue of \( L_{sym} \) and \( u_i \) is its corresponding eigenvector, and index \( i = 1, 2, \cdots, N \).

Step 6: Arrange eigenvalues and eigenvectors in ascending order, so that \( \lambda_i \) is the \( i \)-th smallest eigenvalue and \( u_i \) is its corresponding eigenvector.

Step 7: Arrange \( K \) eigenvectors \( u_1, u_2, \cdots, u_K \) into matrix \( U \in \mathbb{R}^{N \times K} \):

\[
U = \begin{bmatrix} u_1 & u_2 & \cdots & u_K \end{bmatrix}
\]

Step 8: Divide the \( N \) row vectors \( \{y_n\}_{n=1, 2, \cdots, N} \) of matrix \( U \) into \( K \) clusters, using k-means clustering algorithm.

Step 9: Output the results of k-means clustering as the results of spectral clustering.

2.3. Monitoring Thresholds Pre-setting

The pre-setting of thresholds, including density and frequency threshold, originates from different requirements of monitoring intensities in different scenarios [12]. In other words, the monitoring
thresholds are parameters for manual adjustment on demand, which directly influence on the monitoring densities and frequencies in each monitoring block.

2.4. Monitoring Intensity Determination
Monitoring intensities mainly include monitoring densities and frequencies, the former is defined as the portion of monitoring nodes and the latter decides sampling intervals. Concretely, given the environmental function \( f(p, q, t) \), the monitoring density of block \( k \) at time \( t \) denoted as \( \rho_k(t) \) is:

\[
\rho_k(t) = \frac{1}{b-a} \left( \frac{1}{N_k} \sum_{n=1}^{N_k} f(p_n, q_n, t) \right) - a \cdot \zeta_k
\]  

Where \( (p_n, q_n) \) is the position of node \( n \) at time \( t \), \( \zeta_k \) is the density threshold of block \( k \), \( N_k \) is the number of nodes of block \( k \), \( a \) and \( b \) are extremums of \( f(p_n, q_n, t) \):

\[
[a, b] = \left[ \min \frac{1}{N_k} \sum_{n=1}^{N_k} f(p_n, q_n, t), \max \frac{1}{N_k} \sum_{n=1}^{N_k} f(p_n, q_n, t) \right]
\]  

Similarly, the monitoring frequencies of block \( k \) at time \( t \) denoted as \( \text{freq}_k(t) \) is:

\[
\text{freq}_k(t) = \frac{1}{b-a} \left( \frac{1}{N_k} \sum_{n=1}^{N_k} f(p_n, q_n, t) \right) - a \cdot f_0(k) \]  

Where \( f_0(k) \) is the frequency threshold of block \( k \).

2.5. Expected Monitoring Effect
After determining the density and frequency thresholds for each block, the multi-terminal monitoring systems will randomly select several monitoring nodes and communication nodes \([13][14]\). The number of monitoring nodes of block \( k \) at time \( t \) required denoted as \( \hat{N}_k(t) \) is:

\[
\hat{N}_k(t) = N_k \cdot \rho_k(t)
\]  

The number of communication nodes depends on the distance between nodes and communication radius. The expected monitoring effect is shown in figure 1:

**Figure 1. Schematic Diagram of Expected Monitoring Effect**

The nodes in all blocks are divided into three categories: monitoring nodes, communication nodes, and idle nodes. Monitoring nodes are responsible for data collection, communication node is responsible for data transmission, and idle nodes wait for instruction in low power consumption state.
Three types of nodes in each block work together to complete the monitoring tasks for the monitoring systems.

3. Experiments

A large number of experiments have been carried out, including terminal distribution simulation, spectral clustering, data generation, thresholds pre-setting, and monitoring intensity determination, to see experimental monitoring effect of the proposed mechanism.

3.1. Terminal Distribution Simulation

The multi-terminal monitoring systems with different number of nodes, including 100, 200, 500, 1000, 2000, 5000, are simulated. Figure 2 shows the topologies of 500 and 2000 nodes of IoT systems.

![Figure 2. Terminal Distribution Topologies of IoT Systems with Different Sizes](image)

After the topologies of multi-terminal monitoring systems are generated, the similarity matrices of each system can be computed, which can be used for spectral clustering.

3.2. Spectral Clustering

In the experiments, the multi-terminal monitoring systems are divided into different numbers of blocks. The results of spectral clustering of 500 nodes are shown in figure 3:

![Figure 3. Spectral Clustering of 500 Nodes with Different Number of Clusters.](image)
The figure above shows the influence of clustering number, from 2, 4, 6, 8, 10, on the spectral clustering. In order to enhance the persuasiveness of the experiments, the topologies of multi-terminal monitoring systems with 2000 nodes are also tested in the similar way, and figure 4 shows the results.

![Spectral Clustering of 2000 Nodes with Different Number of Clusters.](image)

**Figure 4.** Spectral Clustering of 2000 Nodes with Different Number of Clusters.

It can be seen that the blocks are divided appropriately according to the spatial distribution of nodes.

3.3. Data Generation and Thresholds Pre-setting

In order to reveal the impact of the changes of monitoring environments, several time-varying functions $f(p,q,t)$ are chosen in the experiments. The contour of $f(p,q,t)$ are shown in figure 5.

![Contour of $f(p,q,t)$ at Different Times.](image)

**Figure 5.** Contour of $f(p,q,t)$ at Different Times.
In the experiment, the warm colour area represents the large value of $f(p,q,t)$, the contours of \[ f(p,q,t) \] at selected 6 time points, $t_0, t_1, \cdots, t_5$, indicate that the maximum point of $f(p,q,t)$ moves towards a certain direction.

### 3.4. Monitoring Intensity Determination

In order to show the influence of environment on the monitoring intensity of each block, an instance topology with 2000 nodes is chosen as the clustering object. After finishing the spectral clustering, the sequence numbers of 20 blocks are marked in Figure 6.

![Figure 6. Terminal Distribution Topologies with IoT Systems of Different Sizes](image)

Then the number of monitoring nodes in 20 blocks at different time points are counted, and three blocks with the largest number of monitoring nodes are marked on the bar chart in Figure 7.

![Figure 7. Monitoring Densities of all Clusters at Different Times.](image)

Analysing with reference to the time-varying functions, it is revealed that the change of environment affects the change of key monitoring blocks in the same way.

Comparing with the labels on the bar chart, it shows that monitoring blocks with high intensities moves the same direction with the change of monitoring environment.
3.5. **Experimental Monitoring Effect**

The experimental monitoring effect is presented from two parts, the influence of thresholds on monitoring status and the change of monitoring environments.

3.5.1. **Influence of Thresholds.** The values of thresholds affect the monitoring intensities of the whole multi-terminal monitoring system to a great extent, larger thresholds will definitely increase the monitoring intensities.

The specific impact of the setting of thresholds is shown in figure 8, where the thresholds increase from 0.2 to 1.0 at interval of 0.2. The simulation results match the expectations well.

In fact, the thresholds can be adjusted according to demands of the oriented application scenarios. For critical monitoring areas, the monitoring thresholds should be set larger.

![Spectral Clustering of 2000 Nodes with Different Number of Clusters.](image)

**Figure 8.** Spectral Clustering of 2000 Nodes with Different Number of Clusters.

The most appropriate values of monitoring thresholds generally require trials and errors, and the actual values of thresholds should be set slightly higher than the theoretical values so as to alleviate the accidental failures of terminals.

3.5.2. **Change of Monitoring Environments.** When the monitoring environment changes, the terminal system will automatically adjust the numbers of monitoring nodes and corresponding monitoring frequencies of each blocks. The influences of the change of monitoring environments are shown in figure 9.
Figure 9. Spectral Clustering of 2000 Nodes with Different Number of Clusters.

Comparing with the simulation of monitoring environments in figure 5 and the monitoring densities of all clusters at different times in figure 7, it can be seen that the monitoring intensities adaptively changes according to the current monitoring environment.

4. Conclusions
The cooperative block-variant monitoring mechanism based on spectral clustering solves the problems of rigid monitoring intensities and fixed task roles in traditional multi-terminal monitoring systems. Instead, the proposed mechanism possesses the capabilities of adjusting the monitoring densities and frequencies of each block adaptively. Besides, the overall base of monitoring intensities can be changed flexibly with monitoring thresholds on demand of the concrete application scenarios. The characteristics mentioned above helps to improve the monitoring efficiency and reduce the power consumptions of multi-terminal monitoring systems.

Acknowledgments
This research was supported by the National key Research and Development Program 2018YFB2100100. The authors also grateful acknowledge the supports from colleagues of comprehensive management business department of The Smart City Research Institute of China Electronics Technology Group Corporation.

References
[1] Luxburg U V. A tutorial on spectral clustering[J]. Statistics and computing, 2007, 17(4):395-416.
[2] Wen Jie,Xu Yong,Liu Hong. Incomplete Multiview Spectral Clustering With Adaptive Graph Learning.[J]. IEEE transactions on cybernetics,2020,50(4).
[3] P. Favati,G. Lotti,O. Menchi,F. Romani. Construction of the similarity matrix for the spectral clustering method: Numerical experiments[J]. Journal of Computational and Applied Mathematics,2020,375.
[4] Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(11): 2765-2781.
[5] Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(1): 171-184.
[6] Yuan M D, Feng D Z, Shi Y, et al. Dimensionality reduction by collaborative preserving Fisher discriminant analysis[J]. Neurocomputing, 2019, 356: 228-243.

[7] X. Liu, J. Chen, Y. Wu, Y. Cui, Z. Ding and S. Yang, "An Optimized Residual Network with Block-soft Clustering for Road Extraction from Remote Sensing Imagery," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019, pp. 2767-2772.

[8] Yuan M D, Feng D Z, Liu W J, et al. Enhanced regularized least square based discriminative projections for feature extraction[J]. Signal Processing, 2017, 139: 182-189.

[9] J. Chen, X. Liu, C. Liu, Y. Yang, S. Yang and Z. Zhang, "A Modified Convolutional Neural Network with Transfer Learning for Road Extraction from Remote Sensing Imagery," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 4263-4267.

[10] Yuan M D, Feng D Z, Liu W J, et al. Enhanced regularized least square based discriminative projections for feature extraction[J]. Signal Processing, 2017, 139: 182-189.

[11] Camila P.S. Tautenhain,Mariá C.V. Nascimento. An ensemble based on a bi-objective evolutionary spectral algorithm for graph clustering[J]. Expert Systems With Applications,2020,141.

[12] Xiaojun Yang,Weizhong Yu,Rong Wang,Guohao Zhang,Feiping Nie. Fast spectral clustering learning with hierarchical bipartite graph for large-scale data[J]. Pattern Recognition Letters,2020,130.

[13] Zhanxuan Hu,Feiping Nie,Wei Chang,Shuzheng Hao,Rong Wang,Xuelong Li. Multi-view spectral clustering via sparse graph learning[J]. Neurocomputing,2020,384.

[14] John Christopher R,Watson David,Barnes Michael R,Pitzalis Costantino,Lewis Myles J. Spectrum: fast density-aware spectral clustering for single and multi-omic data.[J]. Bioinformatics (Oxford, England),2020,36(4).